APPX LIBRARY

System Management Manual

APPX 4.x, 2016

> Written for APPX 4.x. Menus, screens and options in current releases may differ.

APPX Software, Inc.

Made on October 10, 2026 from the pages of the APPX Library, https://wiki.appx.com/manuals/library/system-management/. The pages are the current version.

Contents

Chapter 1 **Orientation to the Manual** 8
APPX Main Menu 8
Related Manuals 9
Topics Covered in This Manual 9
Minimal Startup Operation 10
Defining Keystroke Macros 11
Pivot Year 11

Chapter 2 **System Administration** 13
System Administration Menu 13
SYSTEM SETUP
System Setup Menu 14
System Registration 15
Importing Registration Info 17
System Administration File Mgt 21
Differences in Managing Sys Admin Files 22
Estimating Record Counts 23
System Parameters 26
System Parameters Report/Log Overlay 29
Import Pre-3.2 SQL Registration 31
Hook Records 32
Release Information 33
CONFIGURATION
Configuration Menu 33
Default Hierarchy 35
Printers File Maintenance 36
Printers File Maintenance Overlay 37
Forms File Maintenance 39
Forms File Maintenance Overlay 41
File System Groups File Maintenance 43
FMS Controls Overlay for ISAM 45
FMS Controls Overlay for Sybase 46
FMS Controls Overlay for Oracle 48
Table Naming Scheme Hints 50
FMS Controls Overlay for ODBC 52

FMS Controls Overlay for APPX/Net 54
FMS Controls Overlay for DB2 55
Identities 57
Identities Overlay 58
Identities Detail Overlay 59
Disposition Definitions File Maintenance 60

DATABASES/APPLICATIONS

Databases/Applications Menu 63
Applications File Maintenance 65
Applications File Maintenance Overlay 66
Applications Report/Log Defaults Overlay 68
Databases/Applications File Maintenance 69
Databases Report/Log Defaults Overlay 71
Databases File Maintenance Overlay 72
Databases Related Applications Overlay 73
Databases/Applications Language ID Overlay 75
Databases/Applications Report/Log Overlay 76
Design File Management 76
Design File Management Menu 77
Verify Files Menu 80

SECURITY

APPX Security 83
Security Menu 84
Users File Maintenance 85
Users File Maintenance Overlay 87
Users Design Rights Overlay 89
Users Documentation Rights Overlay 90
Users Database Mgt Rights Overlay 91
Users Report/Log Defaults Overlay 93
Review User Keymaps 93
Change Auxiliary Password 94
Users Security Profiles Overlay 95
Users Identities Overlay 96
Security Profiles File Maintenance 98
Security Profiles F/M Overlay 100
Goto Security Code Overlay 102
Set Range Overlay 103

Copy Rights Table Overlay ... 105
Archive Security Codes for All Apps ... 105
Security Profile X-Ref List ... 107
Review System Default Keymaps ... 108
Language Translation ... 109

Chapter 3 **Application Database Management** ... 110
Overview - Database Management ... 110
Database Management Menu ... 110
File Selection ... 113
File Specifications ... 115
File Attributes Overlay for ISAM ... 118
File Attributes Overlay for Sybase ... 118
File Attributes Overlay for Oracle ... 119
File Attributes Overlay for ODBC ... 120
Table Naming Hints ... 122
File Attributes Overlay for APPX/Net ... 123
Optional Path Name Overlay ... 124
Field Overrides Detail Overlay ... 125
Import Files Overlay ... 127
Rebuild Key Files ... 128
Verify Files Menu Overlay ... 128

Chapter 4 **Application Documentation Management** ... 131
'Manage Application Documentation' Menu ... 131
Database Documentation Management ... 132
Select Item Documentation Screen ... 133
Edit Item Documentation Screen ... 134
Documenting Fields ... 136
Process Documentation Management ... 138
Select Process Documentation Screen ... 139
Edit Process Documentation Screen ... 140
External Copy of Documentation ... 141
Documentation Management Utilities ... 142
Items Without Documentation ... 143
Options Without Documentation ... 144

Chapter 5 **Application Security Management** ... 146
'Manage Application Security' Menu ... 146

Database Security Management ... 148
Domains Security ... 149
Files Security ... 151
Fields Security ... 152
Work Fields Security ... 153
Process Security Management ... 155
Process Security for Non-Input ... 156
Process Security for Input ... 157
Security Management Utilities ... 158
Security Lists ... 159
Clear Security Codes ... 160
Archive Security Codes ... 161
Post Security Codes from Archive ... 163

Appendix A **Printing in Unix** ... **164**
Overview - Printing in Unix ... 164
Customizing 'appx_print' ... 164
Disposition Options ... 165

Appendix B **Printing in Microsoft Windows** ... **166**
Overview - Printing in Microsoft Windows ... 166
Customizing 'winprint.exe' ... 166
Keywords in 'winprint.exe' ... 167
Printer/Form Control Options ... 167
Windows Form Names ... 168
Paper Sources ... 170
Printer Driver Issues ... 170

Appendix C **Command Line Options** ... **172**
Standard APPX Command Line Options ... 172
APPX Client Command Line Options ... 174

Appendix D **Creating a Custom APPX Engine** ... **178**
Linking External 'C' Programs into APPX ... 178
Linking External File Mgt Systems into APPX ... 179
Linking an RDBMS into APPX ... 181

Appendix E **APPX Environment Variables** ... **183**
Overview - APPX Environment Variables ... 183
Table E-1: APPX Environment Variables ... 183

GENERAL ENVIRONMENT VARIABLES
APPX/Net Client and APPX/ODBC Variables ... 187
Backward Compatibility Variables ... 187
General Variables ... 190
Import/Export Variables ... 193
KEATERM Configuration Variables ... 193
Presentation Server Variables ... 193
Printing and Job Submission Variables ... 194
Scripting and Regression Testing Variables ... 196
Unix-Based APPX Client Variables ... 197
Windows Variables ... 198
Y2K Option Variables ... 199
DATABASE (RDBMS) INTERFACE VARIABLES
RDBMS Configuration Variables ... 200
RDBMS - ISAM Configuration Variables ... 201
RDBMS - ODBC Variables ... 202
RDBMS - Oracle Variables ... 203
RDBMS - Sybase Variables ... 204
RDBMS - Vision Variables ... 205
DEBUGGING VARIABLES
Debugging - APPX/Net Variables ... 206
Debugging - APPX/ODBC Variables ... 207
Debugging - APPX CGI Interface Variables ... 207
Debugging - CISAM Variables ... 208
Debugging - License Server Variables ... 208
Debugging - Memory Variables ... 209
Debugging - General Variables ... 210
Debugging - Presentation Server Variables ... 213
Debugging - RDBMS Access Variables ... 214
Debugging - Restructure Variables ... 215
Debugging - Security HASP ... 215
Debugging - Language Translation Variables ... 216
Debugging - Unix ... 216
Debugging - Vision Variables ... 216
Debugging - Windows Variables ... 217
Debugging - winprint Variables ... 217
SETTING ENVIRONMENT VARIABLES
Global OS Environment Variables ... 218

Per-user OS Environment Variables 218
Startup Script or Batch Files 219
appx.env Environment Variables 219
Cautions in Setting Environment Variables 220

CHAPTER 1

Orientation to the Manual

APPX Main Menu

This manual describes how to specify and maintain the information that is required for the APPX operational environment before you begin to install or develop applications. It is intended for site installation managers, system administrators, and database administrators of APPX installations.

The manual addresses four major areas: System Administration, Database Management, Documentation Management, and Security Management. Menus for these areas are all accessed from the APPX main menu shown in *Figure 1-1* below.

APPX Main menu

Figure 1-1

Related Manuals

As the APPX system or database administrator, you need to be familiar to some extent with all aspects of the APPX environment, including application design and the run time environment. The APPX *User Manual* describes for end users how to run APPX applications. There is another manual called the *Designer Reference Manual* which describes all of the functions of application design. You should be familiar with these two manuals because they contain references to the system administration function and occasionally direct readers to either the system or database administrator.

In addition, because APPX operates in several different computer environments, the APPX *User Manual* cannot describe every type of computer terminal, keyboard, or printer. You may need to supply manuals or training guides that describe for users how to use your specific computer terminals, keyboards, or printers.

There is also an installation guide that describes how to install APPX on your computer. You need to complete the installation procedure before using any of the screens described in this manual.

Topics Covered in This Manual

This manual encompasses four major components of APPX:

- ***System administration***, including:
- System setup functions
- Configuration management functions
- Database/applications file maintenance and design file management functions
- Security and user file maintenance functions
- Language translation functions (see separate Language Translation manual for details)
- ***Database management*** ***(for a specified database and application)***, including:
- Selecting files for subsequent operations
- Editing file specifications
- Creating files
- Restructuring files
- Initializing files
- Scratching files
- Importing files from other computers
- Exporting files to other computers
- Rebuilding key files
- Verifying file structure and integrity
- ***Documentation management*** ***(for a specified application)***, including:

- Maintaining documentation for data domains, files, and fields
- Maintaining documentation for all types of processes
- Utilities to assist in documentation management
- ***Security management** (for a specified application)*, including:
- Establishing security codes for data domains, files, and fields
- Establishing security codes for all types of processes
- Utilities to assist in security management

Minimal Startup Operation

Chapter 2 describes all of the screens and files that are available in APPX's system administration facilities. You will probably enter information in most of them at some point in time. However, you may want to get started in APPX quickly by providing some minimum information at first, and later return to the system administration screens to supplement the basic information with more complete specifications. Refer to the instructions in Chapter 2 for details of how to perform the minimum required steps that are listed here.

- Select SYSTEM ADMINISTRATION from the APPX main menu and, from here, select SYSTEM SETUP and then SYSTEM ADMINISTRATION FILE MANAGEMENT.
- Select CREATE FILES, and create all of the system administration files.
- Return to the System Administration menu, select DATABASE/APPLICATIONS, and then APPLICATIONS, and enter the name of your first application.
- Return to the Databases/Applications menu, select DATABASES/APPLICATIONS, and enter the name of your first database.
- Select ENTER twice, and enter the database/application combination.
- Return to the System Administration menu, select SYSTEM SETUP, and then SYSTEM ADMINISTRATION FILE MANAGEMENT and run UPDATE SYSDIR.
- Return to the System Administration menu; select SECURITY, and then USERS, and enter names of people who are authorized to use APPX.
- Select DESIGN RIGHTS and enter the applications that each user is authorized to design or maintain or leave blank for all.
- Select DATABASE MANAGEMENT RIGHTS and enter the database IDs that each user is allowed to access or leave blank for all.

At this point, a designer can install or develop the specified application, and run it when it is ready.

Defining Keystroke Macros

> **NOTE: At the present time, *Keystroke Macros* are not implemented for the GUI Client.**

As you become a more experienced APPX system administrator, you may find that as System Administrator, you often perform the same tasks that require the same sequence of keystrokes every time you need to perform them. With APPX's *Keystroke Macro* feature, you can "program" these keystrokes into an option number you select then execute the series of keystrokes anytime just by selecting the option you defined. APPX "remembers" this option and the macro associated with it so it is available for use every time you log in.

To define a macro, begin by pressing the `CTRL` and `R` keys simultaneously (the R stands for *remember).* When you see a message at the bottom of your screen that tells you to *Define Option*, select the option number you want to assign to the series of keystrokes that you will program into the macro. To assign option 77 to the macro, for example, you'll press `OPTION`, `OPTION`, `7`, `7`.

After assigning an option number to the macro, a new message will tell you to *Define Macro*. Now you simply execute your task and APPX will record all of the keystrokes you make. When you are finished (your report, input, or other task is complete), press the `CTRL` and `R` keys again to let APPX know your key sequence is complete. Then any time you want to execute those same keystrokes, just select `OPTION` `77` and the keystrokes will be played back.

Pivot Year

In installations where applications contain date fields that do not include a century component, restructuring files containing such fields, making comparisons to such fields, and calculations on these fields can produce unpredictable results. To ensure that these functions produce the desired results within APPX applications, century components should be added to these date fields or you may define a *pivot year* value.

By defining a pivot year, you direct APPX to determine what century a particular year belongs to. For example, if the pivot year is set to 35, then any date field with a year greater than 35 will be assumed by APPX to have a century value of 19. Any date field with a year less than or equal to the pivot year will be assumed by APPX to have a century value of 20. This pivot year value can be changed as time passes but it is important that it be set to a value best suited to your particular applications and the way they use dates.

The pivot year parameter is set in an APPX *environment variable*. The environment variables file, named *appx.env*, can be found in the APPXPATH directory. Alternatively, there may be a copy in a user's home directory. (On Windows and NT systems, this is the C:\ directory.) If there is a copy of *appx.env* in a user's home directory, its values will override the *appx.env* values in the APPXPATH directory, for all variables found in both places. On UNIX systems, add an "export" statement to your APPX start-up script with the appropriate setting.

`APPX_PIVOT_YEAR=35` (Windows and NT installations) `export APPX_PIVOT_YEAR=35` (UNIX installations)

Alternatively, the APPX_PIVOT_ITEMS environment variable limits pivot logic to masked date items in input processes, without affecting ILF statements. With APPX_PIVOT_ITEMS set to a value between 1 and 99 and with APPX_PIVOT_YEAR not set, pivot year logic will be applied to dates entered with the century component masked out. Pivot year logic will not be applied to those same dates when used with ILF statements such as DATE ADD and DATE BTW. This is unlike the use of the APPX_PIVOT_YEAR variable, which applies pivot logic to both masked date input items and ILF statements. To set APPX_PIVOT_ITEMS, use the same format described for APPX_PIVOT_YEAR.

CHAPTER 2

System Administration

System Administration Menu

You select SYSTEM ADMINISTRATION from the APPX main menu to enter into system administration functions. This calls the System Administration Menu shown in *Figure 2-1* below:

System Administration menu

Figure 2-1

This menu offers five menu options as follows:

- System Setup
- Configuration
- Databases/Applications

- Security
- Language Translation

Each of these menus are described in detail in this chapter.

SYSTEM SETUP

System Setup Menu

The SYSTEM SETUP option on the System Administration menu displays the System Setup menu shown in *Figure 2-2* below.

System Setup menu

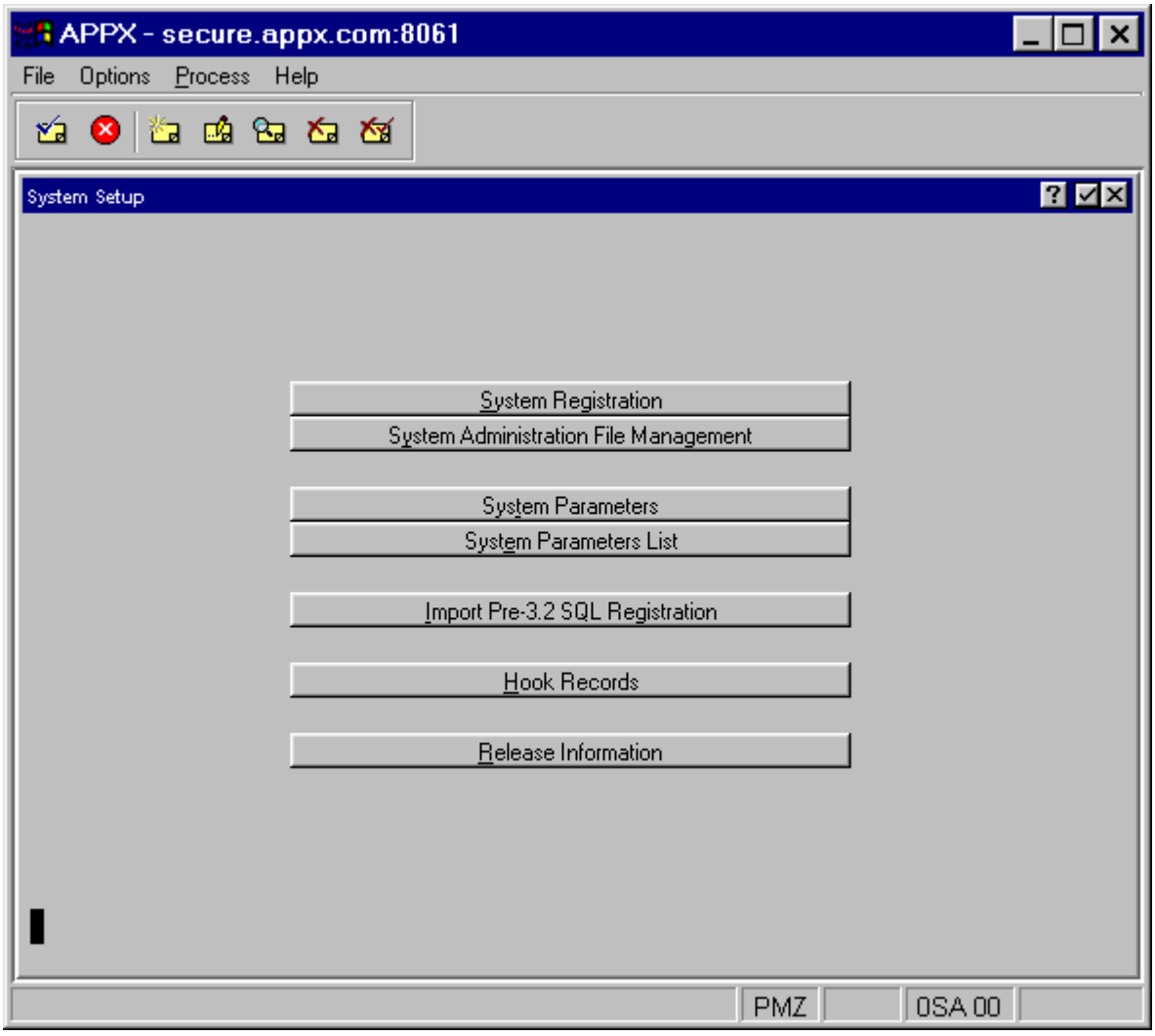


Figure 2-2

Menu options are summarized below and also described in detail in the following sections of this manual:

- ***System Registration*** authorizes the use of APPX and APPX registered applications at your site
- ***System Administration File Management*** provides access to database management functions on System Administration files.
- ***System Parameters*** provides for specification of system-wide default values in the System Parameters file.
- ***System Parameters List*** prints out the complete contents of the System Parameters file.
- ***Import Pre-3.2 SQL Registration*** converts a pre-3.2 SQL registration to satisfy new registration requirements.
- ***Hook Records*** permits specific process names to be overridden at runtime.
- ***Release Information*** displays information pertaining to the release of APPX currently running at your site.

System Registration

The purpose of the registration procedure is to prevent unauthorized use of the APPX utility and application products. In order to validate your system as an authorized site, the system administrator at each site must enter some predefined information into APPX using the System Registration screen shown in *Figure 2-3* below. This screen is accessed by the SYSTEM REGISTRATION option on the System Setup menu.

System Registration screen

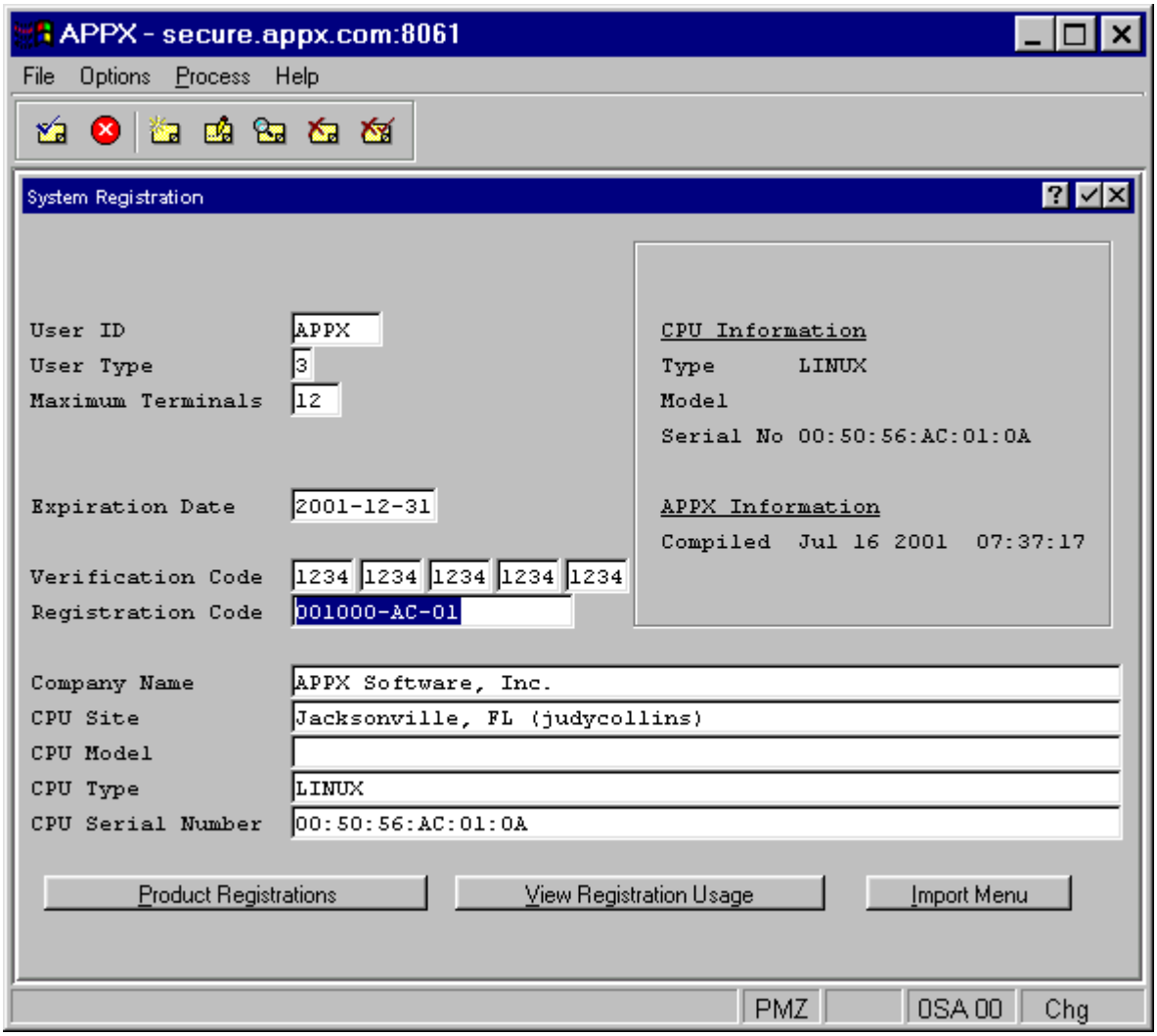


Figure 2-3

Entering Registration Information

In addition to the software media that contains the APPX utility, you should also receive a *System Registration Document* that contains information such as your company name and CPU site.

When you first run APPX after completing the installation procedures, the System Registration screen appears. This screen contains the exact same fields as the *System Registration Document* in the same sequence. You **must** enter the information from the document into the corresponding screen fields **precisely as printed in the document**, taking care to match upper and lowercase letters, including punctuation and embedded spaces if applicable. If you entered the information accurately, you will be able to access APPX. If the information is entered incorrectly, you will receive an error message directing you to review and correct the information. You cannot access APPX until the registration information is correctly entered.

Individual *product registrations* must also be entered. After entering the primary registration information, select the PRODUCT REGISTRATIONS button at the bottom of the screen and enter the product registration information **precisely as printed on your registration document.** Press ENTER after each entry then select END from the title bar to exit.

Registration usage can be viewed at any time by selecting the VIEW REGISTRATION USAGE button at the bottom of the screen. This shows users currently logged in and using APPX. Select END to exit.

Modifying Registration Information

You can select the REGISTRATION option to access the System Registration screen if you need to modify the system registration information, such as the CPU site or company name. You will always receive a new document specifying the modified information. Never enter or modify information unless you receive a new *System Registration Document*.

Importing Registration Info

You can also import registration information directly from an email message that contains your registration document. Select IMPORT MENU from the System Registration Screen to access the Import Registration Menu shown in *Figure 2-4* below.

Import Registration menu

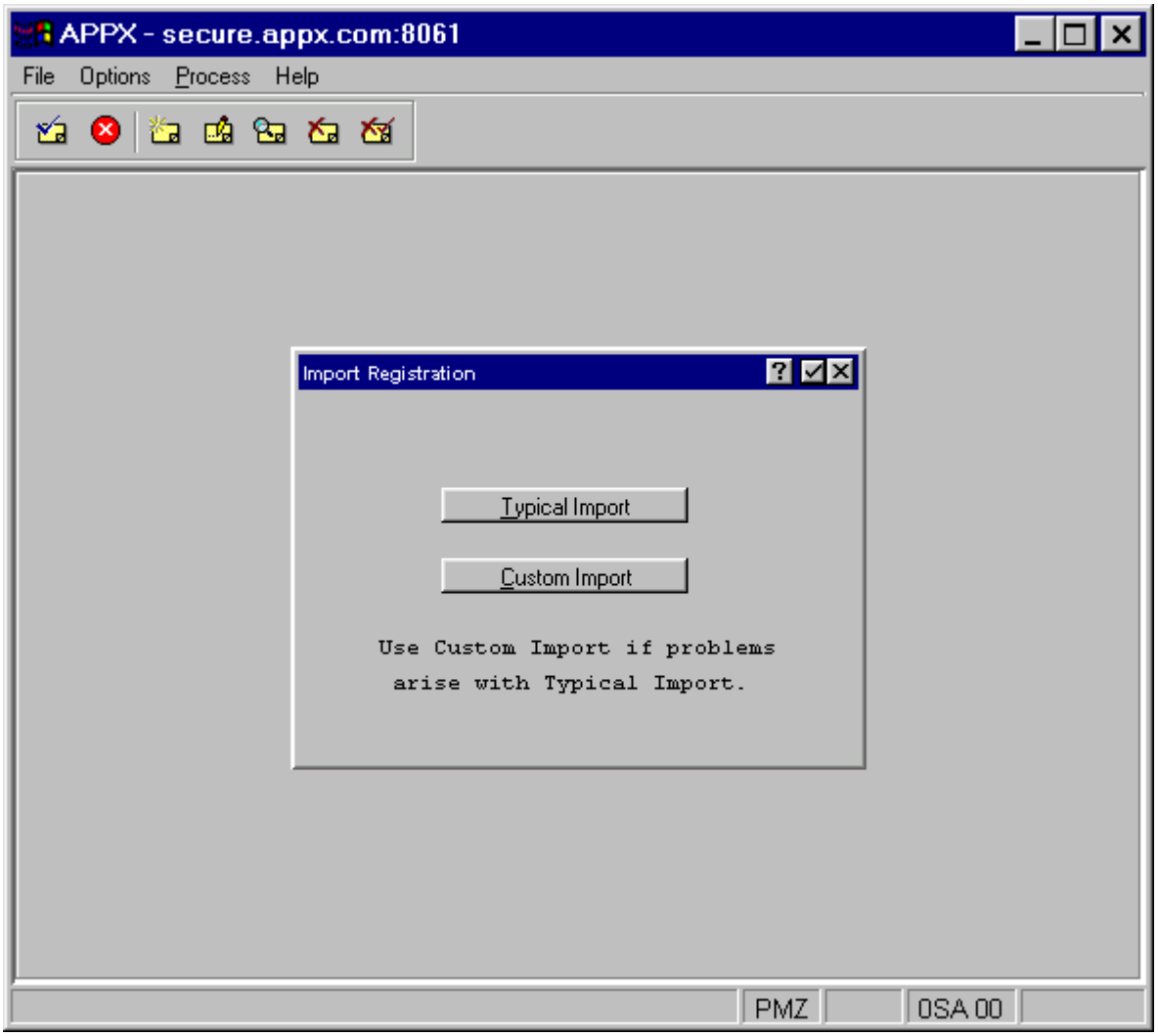


Figure 2-4

This menu contains two options, Typical Import and Custom Import.

Typical Import

This option imports the registration emailed to you by APPX or your supporting distributor in a single step. When selecting this option, you should enter the complete pathname of your saved email message as shown in *Figure 2-5*. Then press ENTER to import.

Figure 2-5. Typical Import Option Overlay

Custom Import

This option provides the same functionality as a typical import but is broken down into steps to give you more feedback regarding the registration import process. Use this option (shown in *Figure 2-6)* if you are having trouble performing a typical import of your registration.

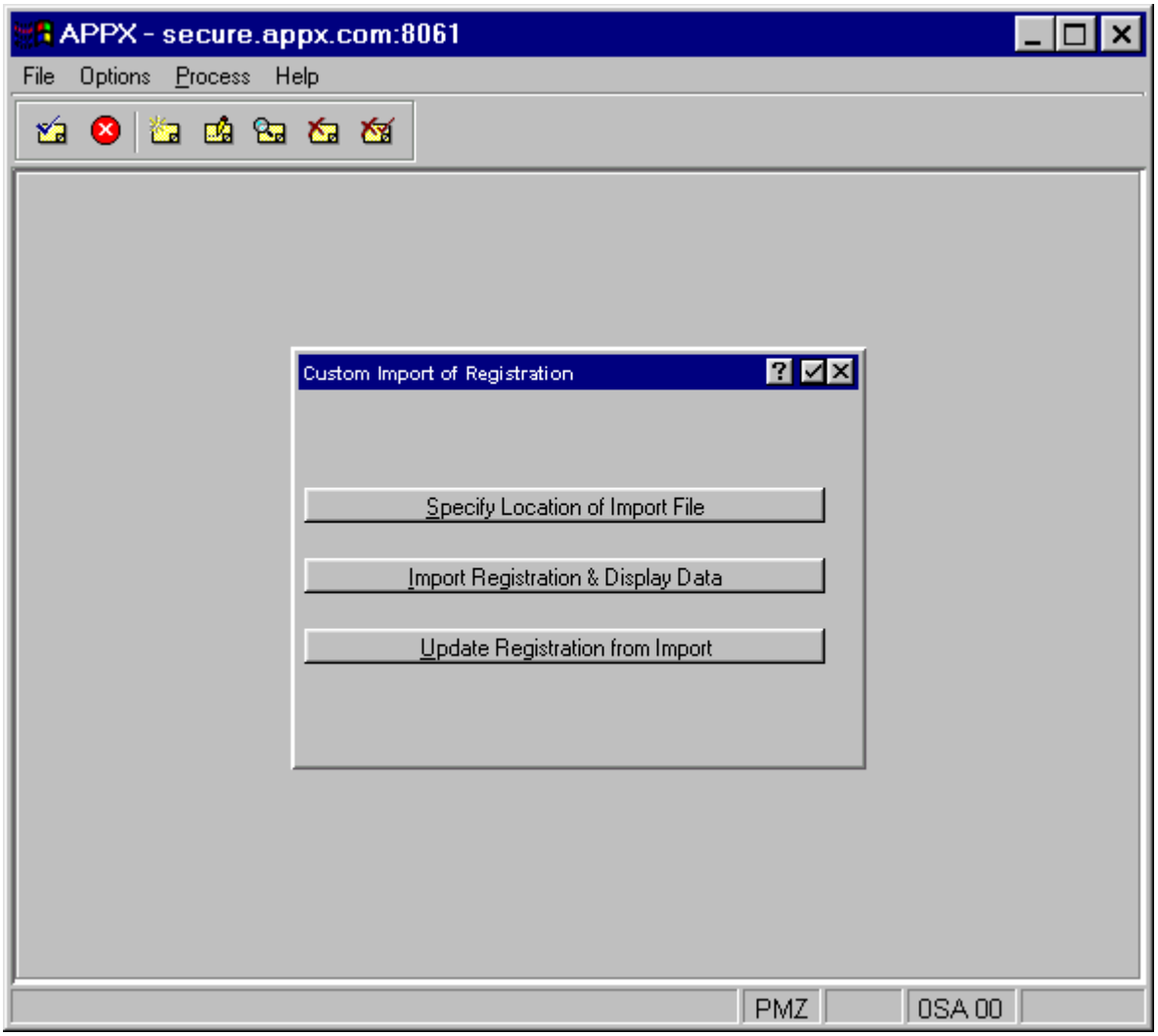


Figure 2-6. Custom Import Option Overlay

To import, select the three options in the order they appear.

Specify Location of Import File

By entering pathname, then press ENTER.

Import Registration & Display Data

To verify that information from the registration document is being imported properly into each registration field. If not, please contact your dealer or APPX support for assistance.

Update Registration from Import

To update registration information. Do not select this option if the display of registration data indicates that registration information is not being imported properly.

System Administration File Mgt

One of the first steps in setting up an APPX installation is to specify information for a special set of files called the system administration files. These files act as a master reference point for all other APPX applications that are run or developed at your installation. Before you can enter information that describes your installation, you need to perform the one-time task of creating the files. The screen you see after selecting the SYSTEM ADMINISTRATION FILE MANAGEMENT option from the System Setup menu is shown in *Figure 2-7* below.

System Administration File Management menu

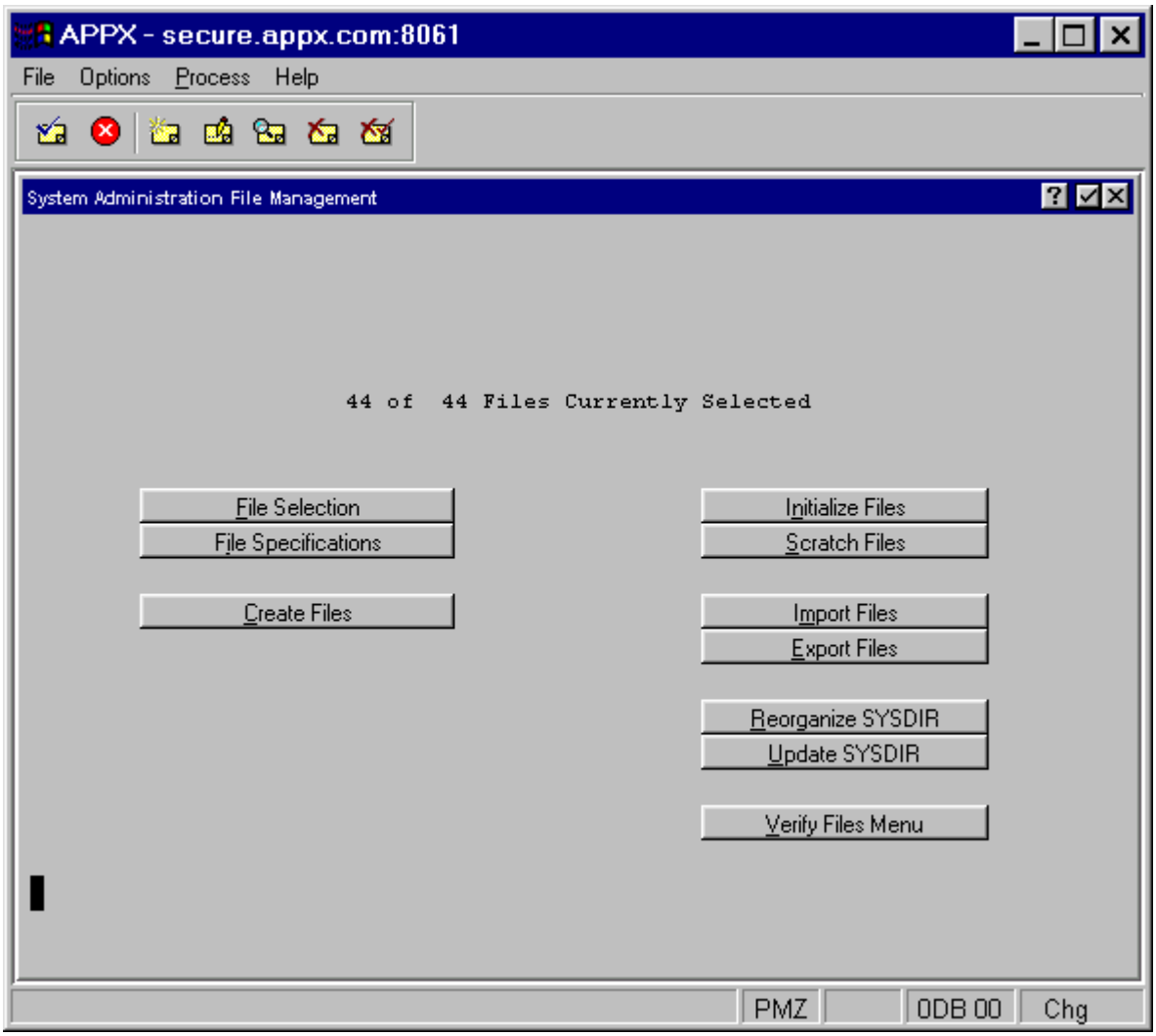


Figure 2-7

A summary of the options available on the System Administration File Management menu is presented below. With the exception of REORGANIZE SYSDIR and UPDATE SYSDIR, similar options are offered for data files on the Database Management menu and are described in detail there. Also see the discussion on differences in treatment of files outlined in the next section and in the overview to Database Management.

- ***File Selection*** allows you to identify which files you want to work on.
- ***File Specifications*** allows you to enter information about each file, including the estimated number of records in each one. Note that these record counts are not used in actually sizing files (except on the DEC/VAX which is no longer supported). Instead, APPX creates files as small as possible and then allocates disk space dynamically as records are added. However, if you enter a value of *0* (zero), APPX will not create the file when you select the CREATE FILES option.
- ***Create Files*** actually creates the files (unless the estimated number of records is zero).
- ***Initialize Files*** sets up selected files with no records by scratching and recreating the files.
- ***Scratch Files*** deletes selected files, eliminating them from the system directory.
- ***Import Files*** and ***Export Files*** allow you to transfer files between two different types of computers.

Reorganize SYSDIR

Will create a new System Directory file called data/0SA/Data/SYSDIRRC.* which should be used to replace the active SYSDIR.* file when all users have exited APPX. Records in the System Directory that are no longer applicable are discarded, as well as references to applications and/or databases that no longer exist.

Update SYSDIR

Will update the System Directory file with applications records and database records. The most common use for this job is updating the System Directory after adding the application and database information for newly installed applications.

- ***Verify Files Menu*** contains utilities used to verify the structure and integrity of APPX files.

Differences in Managing Sys Admin Files

In APPX, the procedures for creating and managing the system administration files are virtually the same as they are for managing data files. Data files store the records that are associated with applications such as Payroll, Order Processing, and Inventory. There are very few differences between the two sets of screens and the information you enter for them. Specific differences are summarized here:

- Database Management offers an additional option to restructure any files in an application, if changes in the structure of the records warrant it. The RESTRUCTURE option is not available on the System Administration File Management menu because the structure of the system administration files is fixed and not subject to change.

- The names of the system administration files are predefined and cannot be changed, whereas designers specify the names of data files in the data dictionary.
- From the System Administration File Management menu, you can reorganize and update your System Directory record in one step.

Estimating Record Counts

Before you create the system administration files, you may provide an estimated record count for each file. Note that these record counts are not used in actually sizing files (except on the DEC/VAX which is no longer supported). Instead, APPX creates files as small as possible and then allocates disk space dynamically as records are added. However, if you enter a value of *0* (zero), APPX will not create the file when you select the CREATE FILES option.

> Note that the files **MAPDEF**, **MAPKEY**, **MAPTYPE**, **TERMKEY**, and **TERMTYPE** are required only if you define custom keymaps. If you only use the APPX-supplied default keymaps, you may specify zero records for these files, in which case the files will not be created.

The following is a list and brief description of system administration files:

APPL (Applications)

Contains one record for each application running or under development at your installation.

ATRIBUTE (Attributes)

Contains records with extended attributes related to other System Administration files.

DATABASE (Databases)

Contains one record for each unique database at your installation.

DBXAP (Databases/Applications)

Contains one record for each valid combination of databases and applications at your installation. A single database may be used by more than one application.

DESIGNER (Application Designers)

Contains one record for each combination of designer and application. A designer may have rights to design more than one application, and each combination is specified in this file.

DFLTMAP (Default Keymap)

Contains fourteen records, provided as part of the APPX product. These include the Menu Entry and the Data Entry keymaps that are supplied with APPX.

DICTNARY (Dictionary)

Contains a record for each word, phrase, sentence or other component in the “base” language and “translated” languages for language translations that occur at the system-wide level.

DISPDEF (Disposition Definitions)
Contains a record for each disposition definition. These records are defined by both designers and users and contain parameters used in the generation of reports and other types of output.

EDITMSG (Copy of MESSAGE For Editing)
Contains a record for each APPX-generated message, warning, and error.

ENVIPARM (Environment Parameters)
Is a system file used internally by APPX.

ENVIRMST (Environment Variable Master)
Is a system file used internally by APPX.

ENVIRON (Environment Settings)
Is a system file used internally by APPX.

ENVITEXT (Text for 'appx.env' header)
Is a system file used internally by APPX.

FLDMAP (Field Level RDBMS Overrides)
Contains one record for each field that requires name or data type overrides in order to communicate properly with an external RDBMS.

FORM (Form Information)
Contains one record for each unique paper form that you use on your printers, such as standard computer paper or check forms.

HOOK (Hook Override Process Records)
Contains one record for each process override hook defined by application designers.

IMPREG (Import Registration Work Fields)
Is a one-record file containing work fields used when importing registrations.

KEYMAP (Keymap)
Contains seven records for each customized keymap you define if you do not use the default APPX keymap.

LANGUAGE (Language Codes)
Contains a record for each language (native and foreign) being used in APPX applications, including the "base" language and the various "presentation" languages.

LOCK (Product Lock)
Contains product registration information, one record for each registered product.

LOCKIMP (Lock Import for Product Registrations)
Contains a record for each product registration lock being imported into the APPX registration.

LUSR (License Users)
Contains record s for each user/product registration code combination currently in use.

MANAGER (Database Managers)
Contains one record for each combination of database and application. A database manager may have rights to manage more than one application, and each combination is specified in this file.

MAPDEF (Keymap Definition)
Contains one record for each combination of terminal type, user, and keymap.

MAPKEY (Keymap Keys)
Contains approximately 300 records (APPX options) for each type of keymap.

MAPTYPE (Map Types)
Contains one record for each different type of keymap you define.

PASSWD (Identities)
Contains one record for each User ID/password combination that will be used to access an external environment, such as an Oracle database.

PRINTER (Printer Information)
Contains one record for each separate printer that is attached to your computer.

PROFILE (Security Profiles)
Contains one record for each set of privileges that you define in the security component of APPX.

PRTCMNDS (EDF Printer Commands)
Is used only in support of the Extended Documentation Facility.

REGIMP (Registration Import File)
Contains a single record of registration information that can be imported into the APPX Registration File.

REGISTER (System Registration)
Contains one record with system registration information.

SCANPRF (Scan Preferences)
Contains records keyed by users and processes that enable APPX to “remember” a user’s scan preferences (files and key paths) for individual processes.

SECRARCV (Security Codes Archive)
Contains archived security code records.

SQL (SQL Parameters)
Contains one record with parameters for APPX's SQL Report Writer.

SYSDFLT (System Defaults)
Contains one record for each FMS (file management system) group and FMS combination. Refer to your installation manual for allowable FMS values.

SYSDIR (System Directory)
Contains one record for each APPX file for all applications at your site.

SYSDIRRC (System Directory Recreate)
Contains one record for each APPX file for all applications at your site.

SYSPARM (System Parameters)
Contains one record, the system parameters.

TERMKEY (Terminal Keys)
Contains approximately 100 records (keys on a keyboard) for each type of terminal that you specify.

TERMTYPE (Terminal Type)
Contains one record for each type of terminal that you define.

USAGE (License Usage)
Contains a record for each user currently using APPX.

USER (User Information)
Contains one record for each user who is defined in your installation.

XLAT (Translations)
Contains a record for each translation of a language dictionary entry that links an entry in one language to its respective entry in another.

System Parameters

The SYSTEM PARAMETERS option on the System Setup menu allows you to specify a number of default values. Unless overridden on other screens (see discussion of default levels in the Default Hierarchy section), these values apply to all APPX users at your installation. As indicated, APPX incorporates a hierarchy of default levels. System parameters are the final places that are checked if no other levels contain overriding values. The System Parameters screen is shown in *Figure 2-8* below.

System Parameters screen

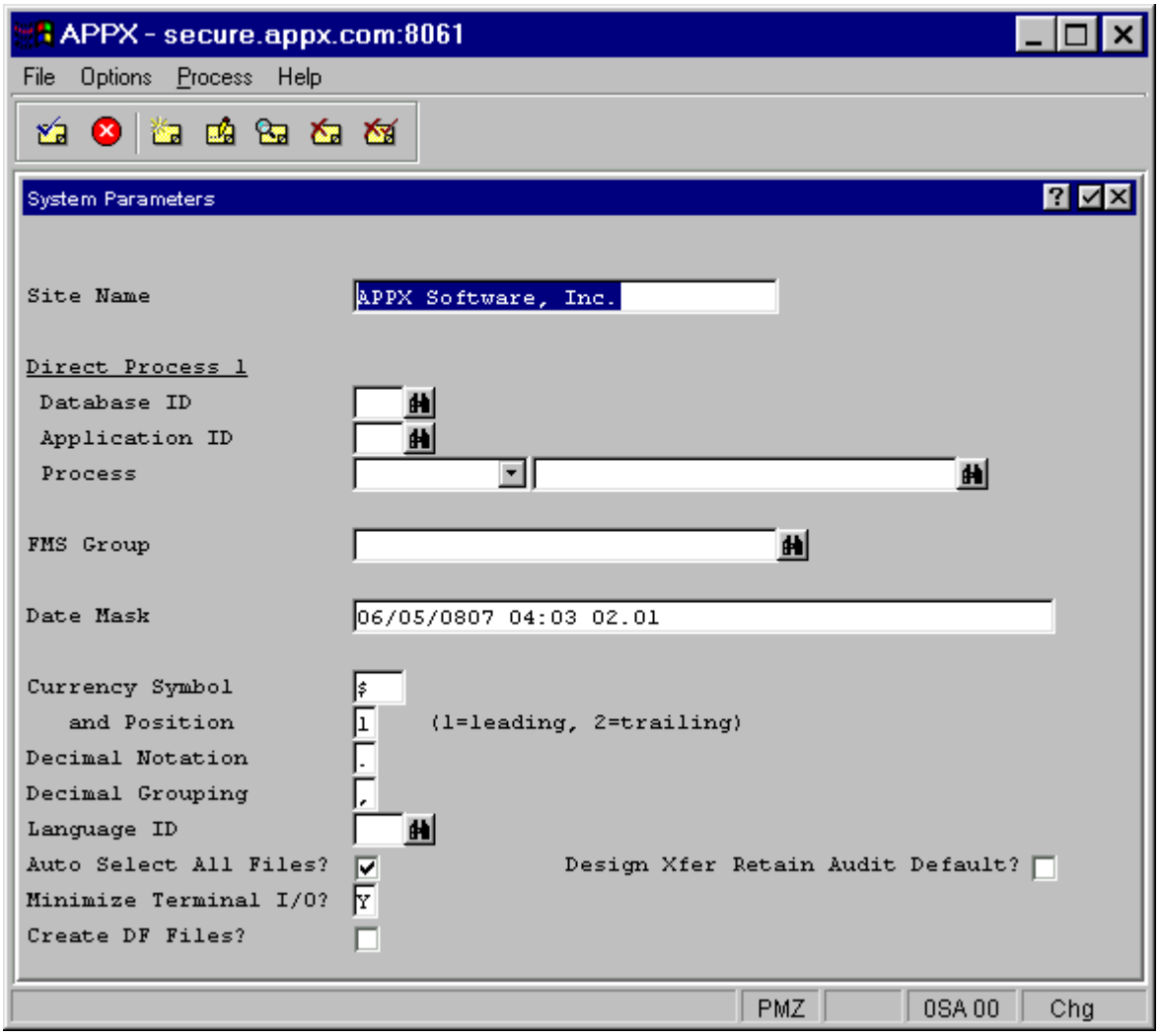


Figure 2-8

The System Parameters screen contains the following fields:

Site Name

Allows you to enter the name of your site, company, or organization.

Direct Process 1

Designates the process that is invoked if a user selects the *Direct Process 1* option while running APPX. Therefore, you should specify the most commonly used menu or other process that users may want to invoke directly, no matter what process they are currently running. The three-part entry consists of:

Database ID

Defines the database where the process is invoked.

Application ID

Defines the application that contains the process that is invoked.

Process Type/Name

Identifies the actual process that is invoked.

FMS Group

Allows you to enter the name of the default File Management System Group at your site.

> **Note:** The next five fields (from *Date Mask* through *Decimal Grouping*) are optional entries. If you do not enter values, APPX provides default values.

Date Mask

Describes the format for displaying dates and times on screens and reports. One example, representing a month-day-year plus hour-minute-second-millisecond sequence mask is: 06-05-0807 04:03:02.01. You must enter all eight parts specified in the date mask. Refer to the Date/Time Display Mask definition in *Chapter 3-14* of the *Designer Reference* manual for more information.

Currency Symbol

Describes the currency symbol (such as *$*) that is in effect at your site.

- ***and Position*** allows you to specify where the currency symbol should appear. Allowable values for this field are ***Leading***, if the currency symbol precedes the numeric values; or ***Trailing***, if the symbol follows the numeric values.

Decimal Notation

Specifies the one-character symbol that separates the integer and fractional parts of money values. For example, in the United States and Canada, the symbol is a period (.).

Decimal Grouping

Specifies the one-character symbol that separates groups of three digits representing thousands, millions, etc. For example, in the United States and Canada, this symbol is a comma (,).

Language ID

Specifies the default presentation language for APPX applications. All applications will be presented in the designated language unless overridden at the database, application or user level. This field should be left blank if the Language Translation feature is not being used. Language codes can be scanned by clicking the *Scan* icon.

Auto Select All Files?

Controls how files are selected when using System Administration File Management, Design File Management, and Database Management. If set to “yes” ☑, all files are automatically selected when you initially enter these screens. If set to “no” ☐, APPX will present you with a list of available files to select from. At this point, you can select some files, all files, or no files. If left as a question mark ?, APPX will select all files.

Design Xfer Retain Audit Default?

Controls the audit dates that are placed into processes, files, and fields which are created by the Design Transfer facility in Application Design. If set to “yes” ☑, APPX will place the audit dates of the parent process, file, or field into the newly created process, file, or field. If set to “no” ☐, APPX will place the current date into the newly created process, file, or field.

Minimize Terminal I/O?

Determines whether APPX should optimize the number of characters to send to the terminal during each screen refresh. Setting this parameter to “yes” or "Y" will improve screen paint rates at the expense of a slight decrease in overall system performance. Setting this parameter to “no” or "N" will improve system performance if your system is heavily loaded, but will decrease screen paint performance. (Note: This field is a one-character alpha field, not a logic field.)

Create DF Files?

If this machine is licensed for the Extended Documentation Facility, select “yes” ☑ in order to create the related Application Design files.

When you complete all of your entries on this screen and press ENTER, the System Parameters Report/Log overlay appears as described in the next section.

System Parameters Report/Log Overlay

After pressing ENTER on the System Parameters screen, the System Parameters Report/Log overlay appears as shown in *Figure 2-9* below. It allows you to specify default values for reports and logs. This same overlay screen appears at four other places in system administration functions, namely, Applications Report/Log Defaults overlay, Databases Report/Log Defaults overlay, Databases/Applications Report/Log Defaults overlay, and Users Report/Log Defaults overlay. Together these provide the information APPX uses to apply the hierarchies described in the Default Hierarchy section.

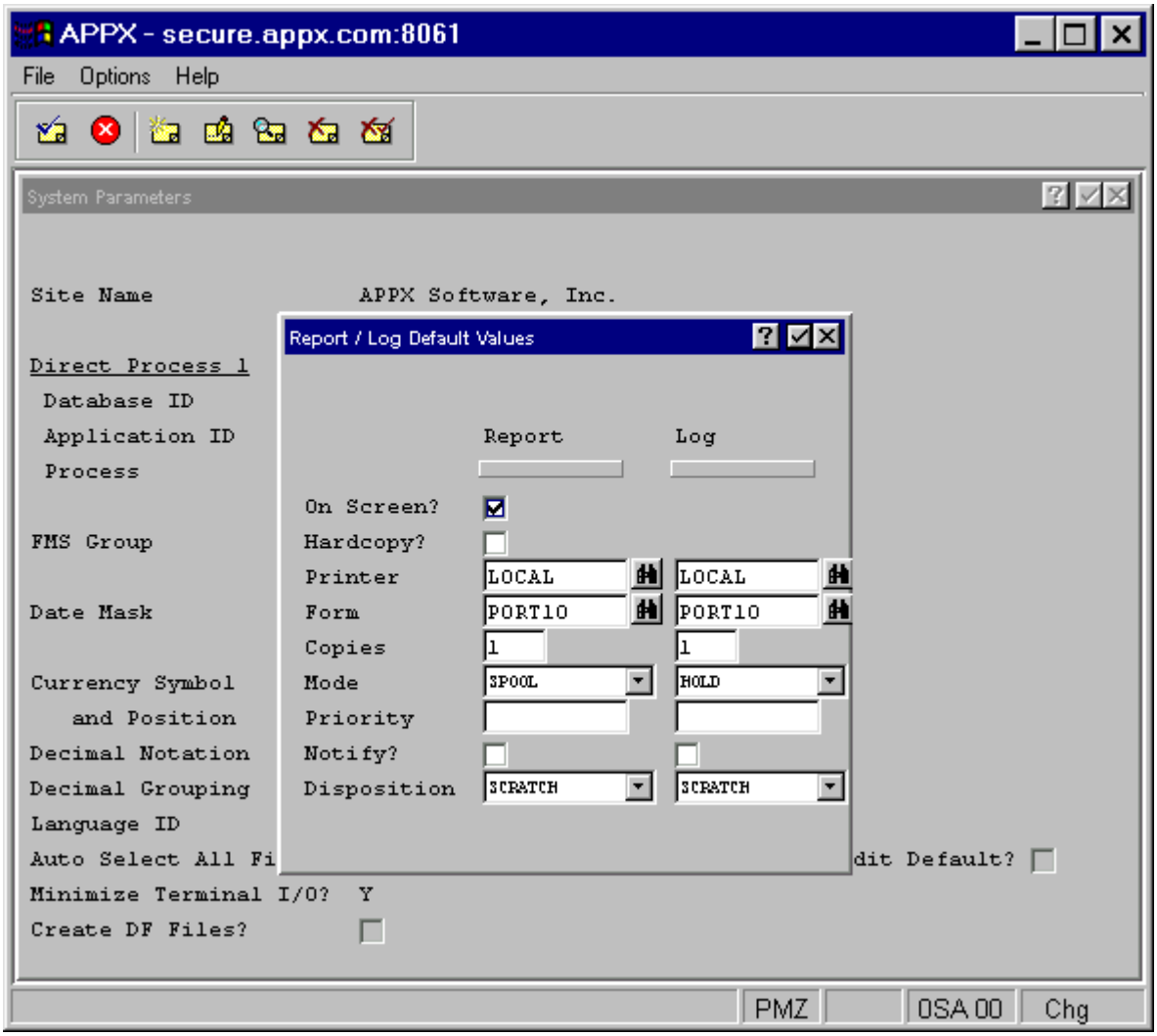


Figure 2-9. System Parameters Report/Log Overlay

The overlay screen is formatted into two columns, one for information about APPX reports and the other for APPX logs (such as an error log). All fields are optional. If all levels of defaults are blank, APPX supplies default values. Some options are not available on some computer systems. You specify the following for both reports and logs:

On Screen?

Select “yes” ☑ or “no” ☐ to indicate if reports are to display on the terminal screen or not. This item is not applicable for logs.

Hardcopy?

Select “yes” ☑ or “no” ☐ to indicate if reports are to be printed or not. This item is not applicable for logs.

Printer
Specifies the default system printer. This is an installation-defined name that corresponds to a system-specific identifier in the printer file.

Form
Specifies the default form number to use for printing. This is also installation-defined, corresponding to a system-specific identifier in the form file.

Copies
Indicates the default number of copies to generate.

Mode
Indicates how print/log files are to be handled. Allowable values for this field are:

- SPOOL spools files to disk and prints according to its priority.
- HOLD spools files to disk and holds until the files are released for printing by an operator.
- KEEP spools files to disk but does not place an entry for them in the print queue.

Priority
Designates the system-specific print code priority.

Notify?
Select "yes" ☑ or "no" ☐ to indicate if the user is to be notified upon completion of printing the report.

Disposition
Indicates the default print disposition. Allowable values for this field are:

- SCRATCH deletes the print file from disk after printing and removes its entry from the print queue.
- SAVE saves the print file on disk after printing, but removes its entry from the print queue.
- REQUEUE saves the print file on disk after printing and places a new entry in the print queue.

Import Pre-3.2 SQL Registration

The IMPORT PRE-3.2 SQL REGISTRATION option on the System Setup menu is provided to satisfy a change in registration requirements. Starting with Release 3.2 of APPX, the APPX SQL Application is protected with an Application Lock. This lock requires a new SQL registration be added to your master APPX registration. This utility takes an existing SQL registration and generates the required Product Registration needed by the new Application Locking mechanism and adds it to your site registration. Before running this conversion, you must copy $APPXPATH/Q00/SQL/Data/PARAM.dat into a new file so that your current SQL registration is used. The name of this new file is $APPXPATH/0SA/Data/SQL.dat.

Hook Records

The HOOK RECORDS option on the System Setup menu permits specific process names to be overridden at run time. If a record exists in this file for a specific process, APPX will invoke the defined override process instead of the original process. This can be done globally for all databases and/or users (by leaving the Database and/or User ID fields blank) or for a specific database and/or user (by specifying a Database and/or User ID). Hook records can be deactivated without deletion by removing the check from the *Active?* checkbox. A sample hook record for overriding an input process is shown in *Figure 2-10.* In this example, any time user RFD selects an option that executes the input process *Customers,* the input process *Customers New* is executed instead. All other users will continue to run the process *Customers* when selected.

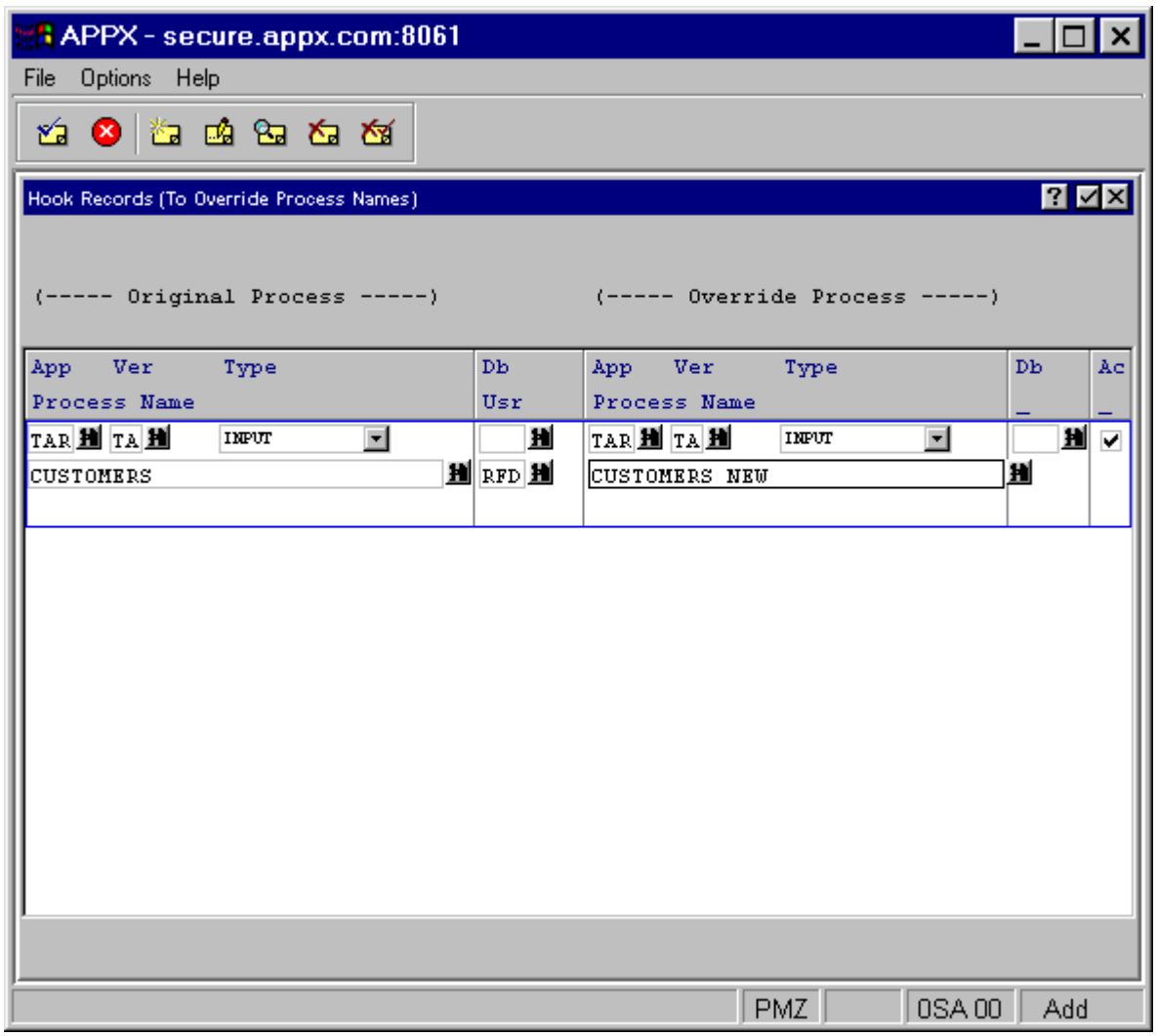


Figure 2-10. Sample Hook Record for Overriding Input Process

Release Information

The RELEASE INFORMATION option on the System Setup menu displays information about the release of APPX currently running as shown in *Figure 2-11*.

Figure 2-11. Release Information

CONFIGURATION

Configuration Menu

The CONFIGURATION option on the System Administration menu displays the Configuration menu, shown in *Figure 2-12* below. Here you can define your specific hardware configuration. APPX operates in a number of different hardware environments, and you need to describe the factors in your particular environment when you first install APPX.

Configuration menu

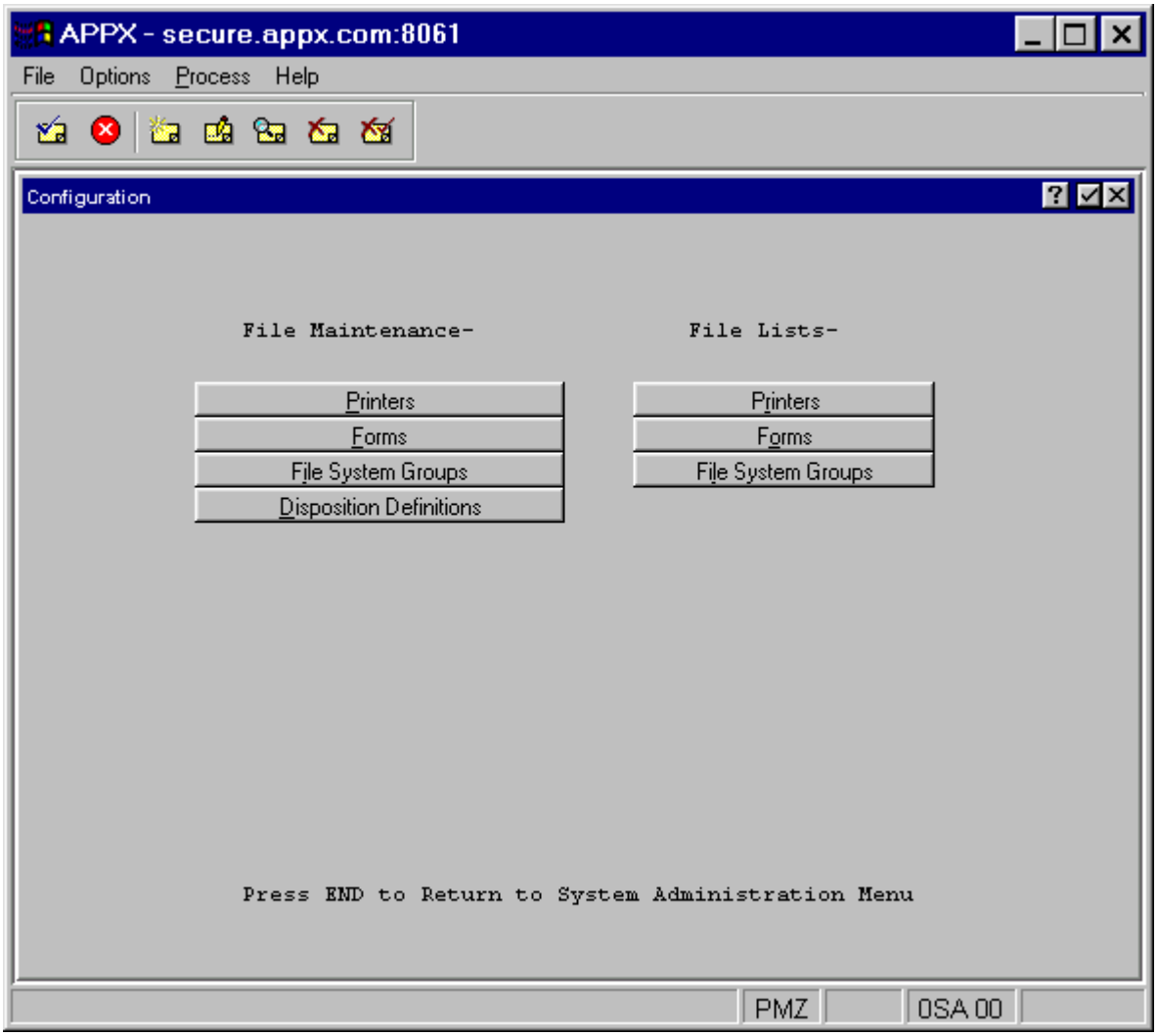


Figure 2-12

File Maintenance

Each of the file maintenance options is described later in this manual. See Printers File Maintenance, Forms File Maintenance, File System Groups File Maintenance, and Disposition Definitions File Maintenance for details.

File Lists

Corresponding file list options provide for printing the complete contents of the configuration files, except for Disposition Definitions.

Default Hierarchy

APPX was designed to allow site-wide standardization and, at the same time, provide designers and users with the flexibility to create a customized environment to work in. This balance is accomplished by a method of default levels that are incorporated into the system. As you describe the components of your system configuration (including printers, forms, terminals, etc.), you begin at a system-wide level. You initially provide default values that are referenced or applied at run time if no further information is provided on subsequent screens. For example, you designate a default printer name at the installation-wide level and, optionally, at a database level, a combined database/application level, and an individual user level. This sequence allows you to provide a site standard, and yet customize the default printer name down to an individual level if so desired.

Two default hierarchies are provided within APPX, one that is in effect during application design and one during run time.

Application Design Default Hierarchy

The following is the default hierarchy for application design:

1. APPX first refers to and uses the values from the User screens. If a value is not entered for a particular component (such as a printer name), then...
2. APPX refers to and uses the values from the Application screens. If a value is not entered here, then...
3. APPX refers to and uses the values from the System Parameters screens. If a value is not entered here, then...
4. APPX provides a value.

Run Time Default Hierarchy

The following is the default hierarchy for run time:

1. APPX first refers to and uses the values from the User screens. If a value is not entered for a particular component (such as a printer name), then...
2. APPX refers to and uses the values from the Database/Application screens. If a value is not entered here, then...
3. APPX refers to and uses the values from the Database screens. If a value is not entered here, then...
4. APPX refers to and uses the values from the System Parameters screens. If a value is not entered here, then...
5. APPX provides a value.

Printers File Maintenance

You use Printers File Maintenance from the Configuration menu to identify and describe the attributes of each printer at your installation. The name you assign to a printer here is the one you enter in other system administration screens when specifying a printer. The Printers File Maintenance screen is shown in *Figure 2-13*.

Printers File Maintenance screen

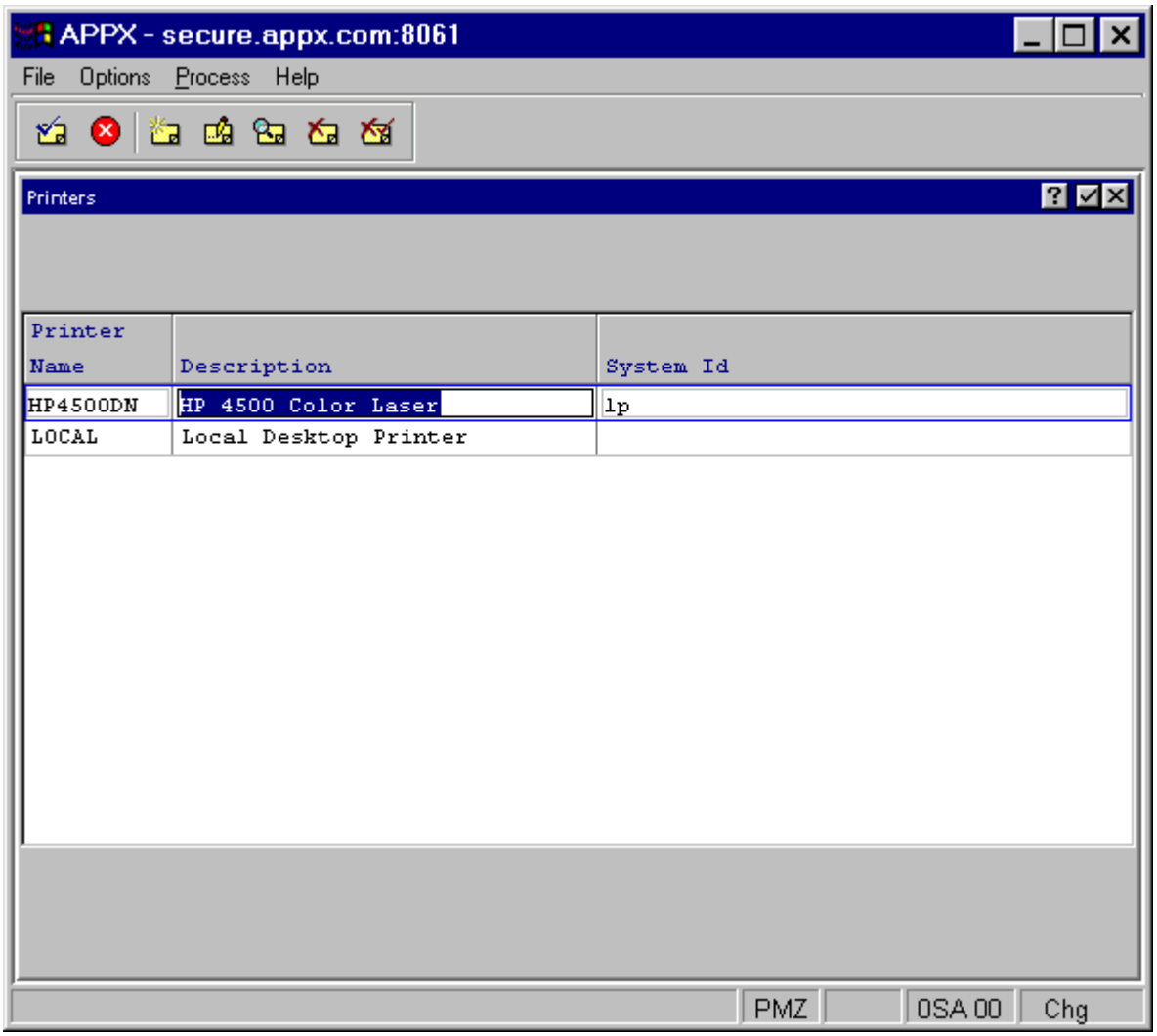


Figure 2-13

The Printers File Maintenance screen contains the following fields:

Printer Name

Specifies a unique identifier for each different printer at your installation. This is the key to the printer file, so each name must be unique. If, for example, you have a low- and a high-speed line printer, and a laser printer, you could assign names of *LINE HIGH*, *LINE LOW*, and *LASER1* (the *1* allows you to add a second and subsequent laser printer later). You can also refer to printers by location, or simply assign them unique numbers.

Description

Allows you to enter a longer description of the printer.

System ID

Identifies the name of the printer as it is defined in Control Panel, Printers on Windows or NT systems. This identifier is both case and spacing sensitive. On Unix and NT systems, this field is not explicitly referenced. However, it is defined as an alternate unique key, therefore a unique value should be entered for it. Be sure to put the hardware device address for your printer in the *Printer Device* field on the continuation overlay.

When you press ENTER from this screen, the Printers File Maintenance overlay described in the next section appears.

Printers File Maintenance Overlay

When you press ENTER from the Printers File Maintenance screen, the Printers File Maintenance overlay displays as shown in *Figure 2-14* below. Here you can enter additional information for the selected printer.

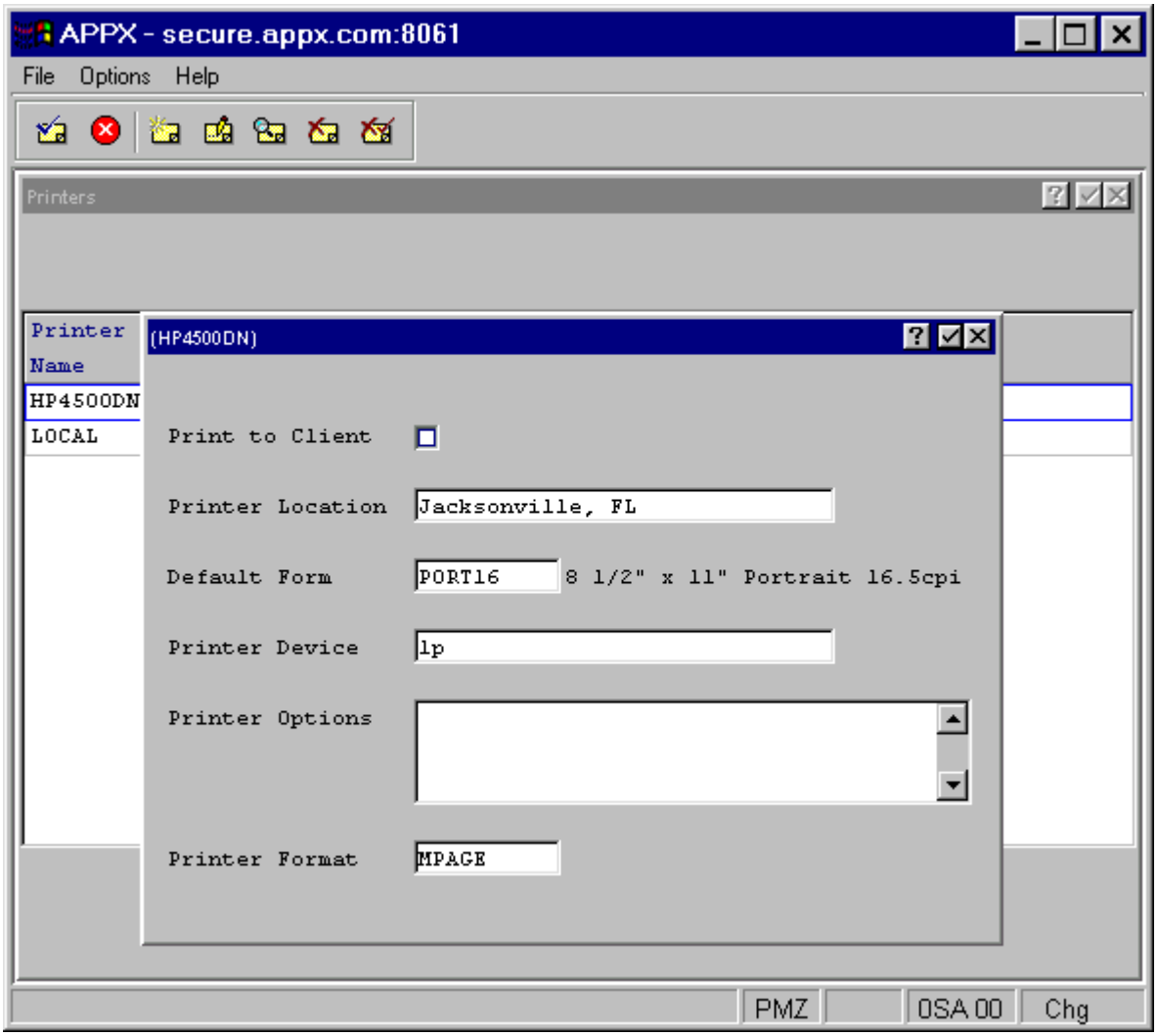


Figure 2-14. Printers File Maintenance Overlay

The Printers File Maintenance overlay contains the following fields.

Print to Client?

Indicates whether or not the document should be printed to the client's printer. Click the checkbox to toggle between "yes" ☑ and "no" ☐.

Printer Location

Specifies the location of the printer. If you have several similar printers at different locations, this information may be an essential means of identification.

Default Form

Specifies the usual form name for this printer.

Printer Device

Specifies the hardware device address for the physical printer.

Printer Options

Allows you to customize printing by passing additional printer-related command line options to the APPX printing utility. In Unix, when a print file is sent out it is stored in the directory */tmp*, or in the directory specified in the *APPX_PRT_FI_DIR* environment variable. When a print file is sent out with a print mode of *HOLD* or *SPOOL*, the file in */tmp* is sent to the *lp* command by using the script *appx_print*. When sending a file to *lp*, APPX appends first the printer options (the string specified here), and then the *Form Control* (found in the Forms file) to the Unix call to *lp*. In Windows and NT, the procedure is similar except that the print file is stored in the directory *%APPXPATH%\print\{user ID}*, and the Windows/NT equivalent to *appx_print* is *winprint.exe*. For an extended discussion of printing in Unix, see Appendix A in this manual. For an extended discussion of printing in Windows or NT, see Appendix B in this manual.

Forms File Maintenance

You use Forms File Maintenance from the Configuration menu to identify and specify the attributes of each standard or customized form that you use at your installation. The name you assign to a form here is the one you enter on other system administration screens when specifying a form. The Forms File Maintenance screen is shown in *Figure 2-15*:

Forms File Maintenance screen

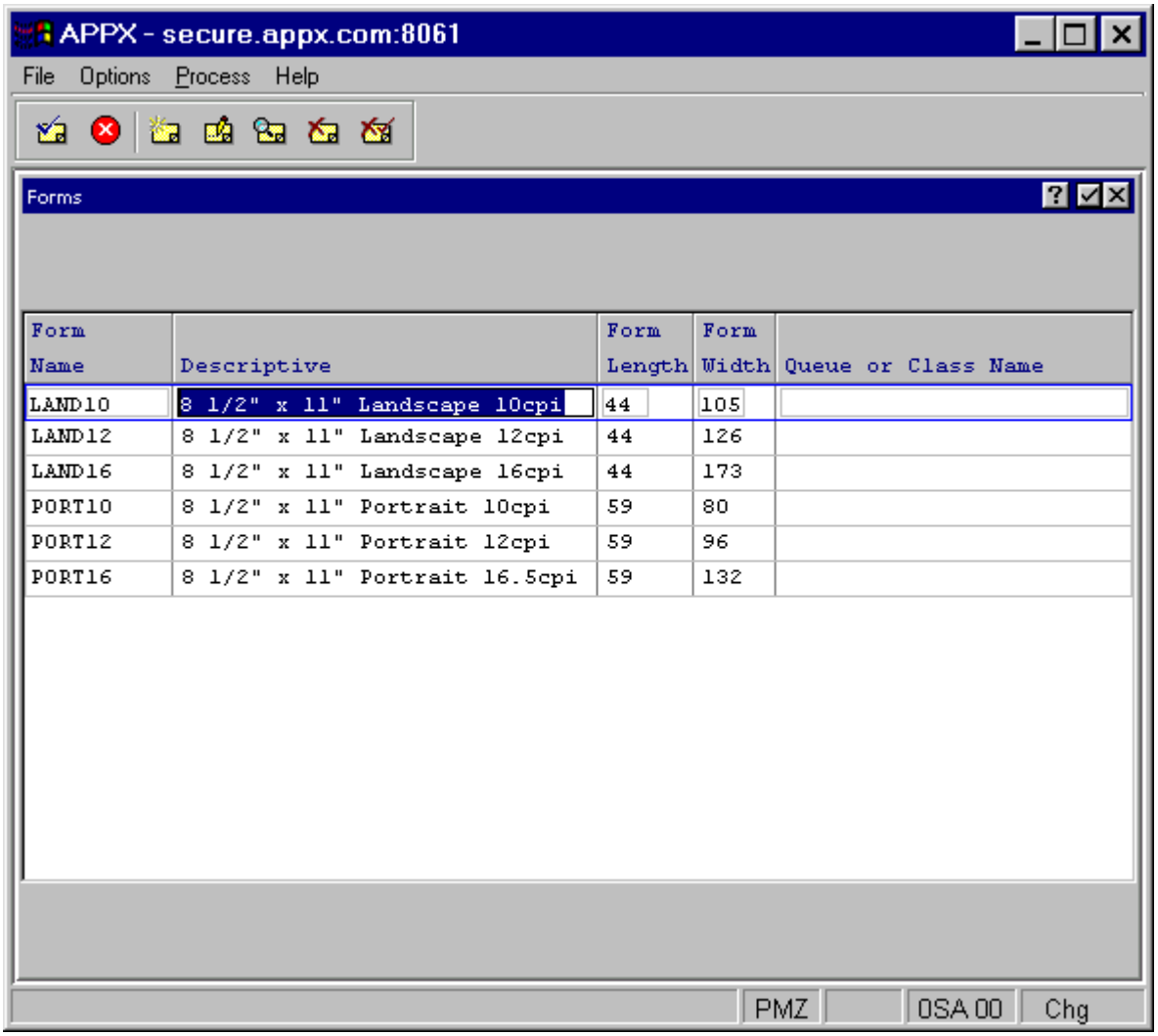


Figure 2-15

The Forms File Maintenance screen contains the following fields:

Form Name

Is a unique identifier for each different form that you use at your installation. This is the key to the form file, so each name must be unique. You may choose to enter a form number, if that is the way you identify forms, or you can enter a more descriptive name, such as *132-COLUMN*.

Description

Allows you to enter a description of the form.

Form Length

Specifies the number of lines that are available for printing on the form. For example, if the form contains 66 lines, but the printer automatically leaves a margin of three blank lines at the top and bottom of each page, enter 60. A value of zero causes continuous printing without page breaks.

Form Width

Specifies the number of columns that are available for printing on the form. For example, if the form contains 85 columns, but the printer automatically leaves three blank columns on the left and two blank columns on the right, then 80 would be entered here.

Queue or Class Name

Controls the print queue to which the report will be submitted on Unix systems. RS/6000s use print queues, so this field therefore may be filled in. HP/9000s do *not* use print queues, so this field therefore should *not* be filled in. For Windows or NT systems, this field is normally left blank.

When you press ENTER from this screen, the Forms File Maintenance overlay described in the next section appears.

Forms File Maintenance Overlay

When you press ENTER on the Forms File Maintenance screen, the Forms File Maintenance overlay appears as shown in *Figure 2-16* below:

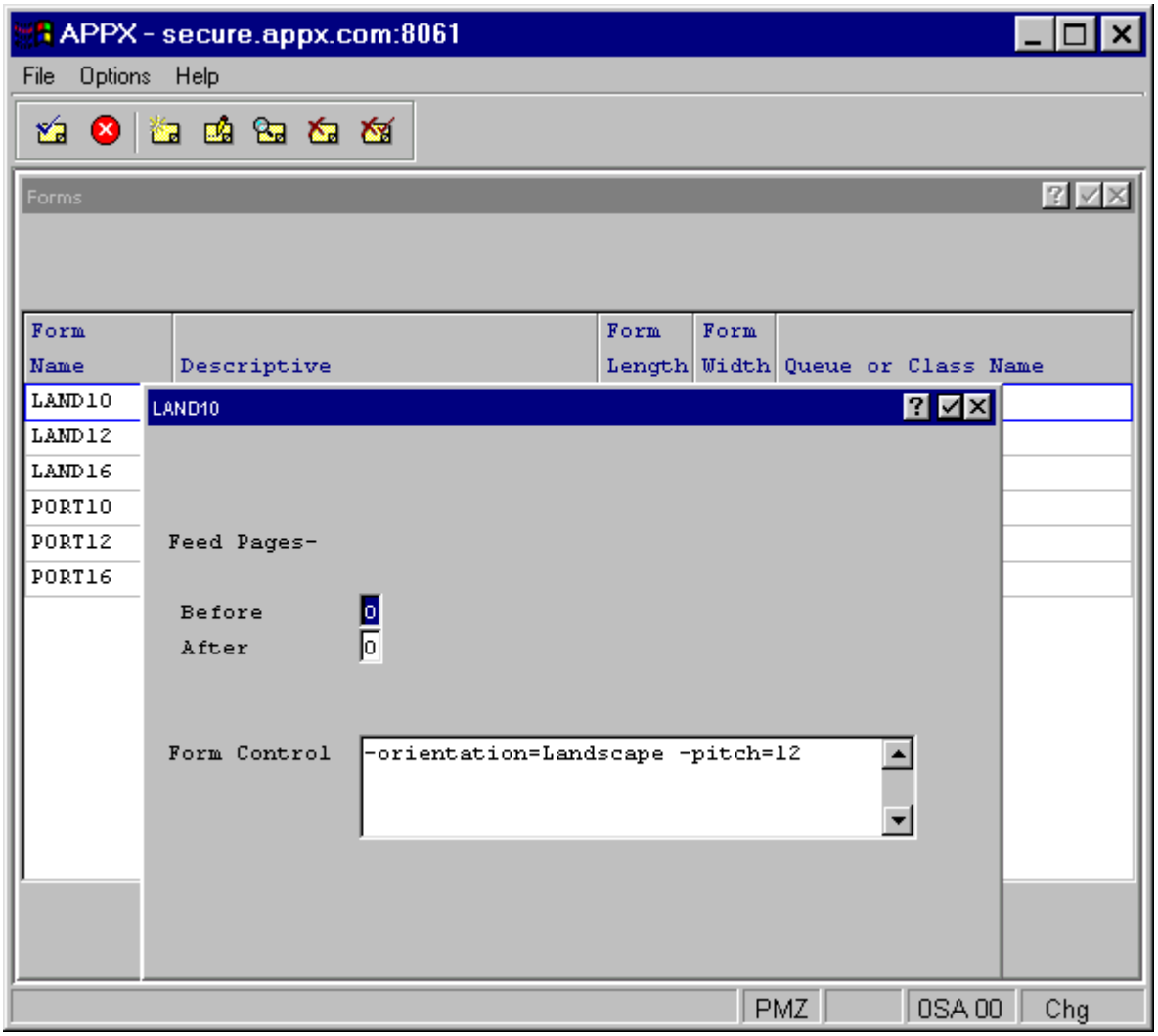


Figure 2-16. Forms File Maintenance Overlay

The Forms File Maintenance overlay contains the following fields:

Feed Pages

Allows you to designate whether you want automatic form feeds to occur at the start or completion of a report using this form.

- Before indicates how many form feeds should occur at the beginning of the report.
- After indicates how many form feeds should occur at the completion of the report.

Form Control

Allows you to customize printing by passing additional forms-related command line options to the APPX printing utility. See the *Printer Options* field on the Printers File Maintenance overlay for more information.

File System Groups File Maintenance

You use File System Groups File Maintenance from the Configuration menu to specify information about individual file group/file management system combinations. The File System Groups File Maintenance screen is shown in *Figure 2-17*.

File System Groups File Maintenance screen

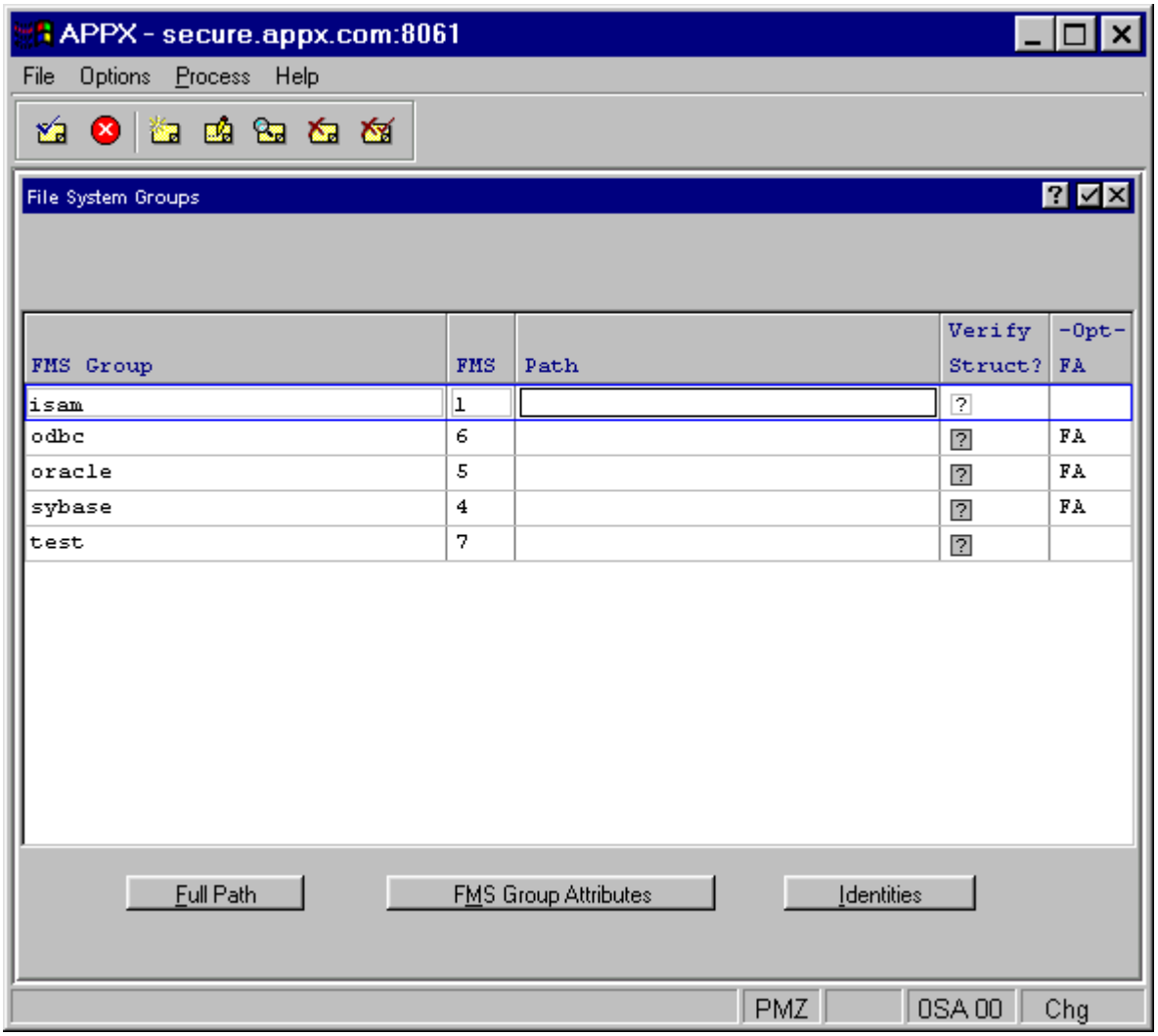


Figure 2-17

The File System Groups File Maintenance screen contains the following fields:

FMS Group

Specifies the name of a file management system (FMS) group. FMS group names are user-defined.

FMS

Specifies the type of ISAM (Indexed-Sequential Access Method) used in the file management system. These values are pre-assigned and computer system specific. Note that some of these interfaces are separately licensed products from their respective vendors and require additional installation files from APPX Software. Valid FMS type values are:

- 1 identifies APPX's default ISAM, known as APPXIO, on all supported Unix and Windows/NT systems.
- 2 identifies ACUCOBOL's Vision ISAM, available on all supported Unix systems.
- 3 identifies Informix's C-ISAM, available on all supported Unix Systems.
- 4 identifies a Sybase database, available on HPUX, AIX, Solaris, Linux, and Windows/NT.
- 5 identifies an Oracle database, available on HPUX, AIX, Solaris, Linux, and Windows/NT.
- 6 identifies an ODBC database, available on Windows/NT systems.
- 7 identifies an APPX/Net arrangement, which can include a mix of systems.
- 8 identifies a DB2 database, available on HPUX, AIX, Solaris, Linux, and Windows/NT.

Path

Allows you to specify a default path name that files in the combined *FMS Group/FMS* will be assigned to.

Verify Structure?

Allows you to specify if APPX should verify whether the structure of any file in this group changed. The verification, if specified, occurs when a file is referenced. Click the checkbox to toggle back and forth between "yes" ☑ and "no" ☐. "Yes" requires the verification process, and "no" does not. In general, after an application/database is in production for a while, you do not need to verify the structure each time. During development when changes are introduced frequently, you may want to verify structures routinely.

The screen also contains these options:

Full Path

Displays the full path name of the FMS group as the scrolling display truncates the path name to fit in the designated column.

FMS Group Attributes

Provides necessary additional information for accessing files for certain FMS types. For example, the database server and database names for RDBMS's such as Oracle and Sybase must be provided. The format of the overlay depends on the FMS Type you have designated for this file system group. See the appropriate section for details about the FMS Controls overlay for ISAM, Sybase, Oracle, ODBC, and APPX/Net.

Identities

Are used to store login information (User ID, realm, and password) for various external interfaces such as an RDBMS. An identity record can exist for each FMS Group/User combination. See the Identities section for an explanation of how identities are used and the hierarchy in which they are employed. See the Identities Overlay for a selected FMS group for specific information. A similar Identities Overlay for a selected user is available in Users File Maintenance. Both overlays provide access to the same login information.

FMS Controls Overlay for ISAM

If you select the FMS GROUP ATTRIBUTES option in File System Groups File Maintenance for a file system group with FMS types 1, 2, or 3, the FMS Controls overlay for ISAM appears as shown in *Figure 2-18* below.

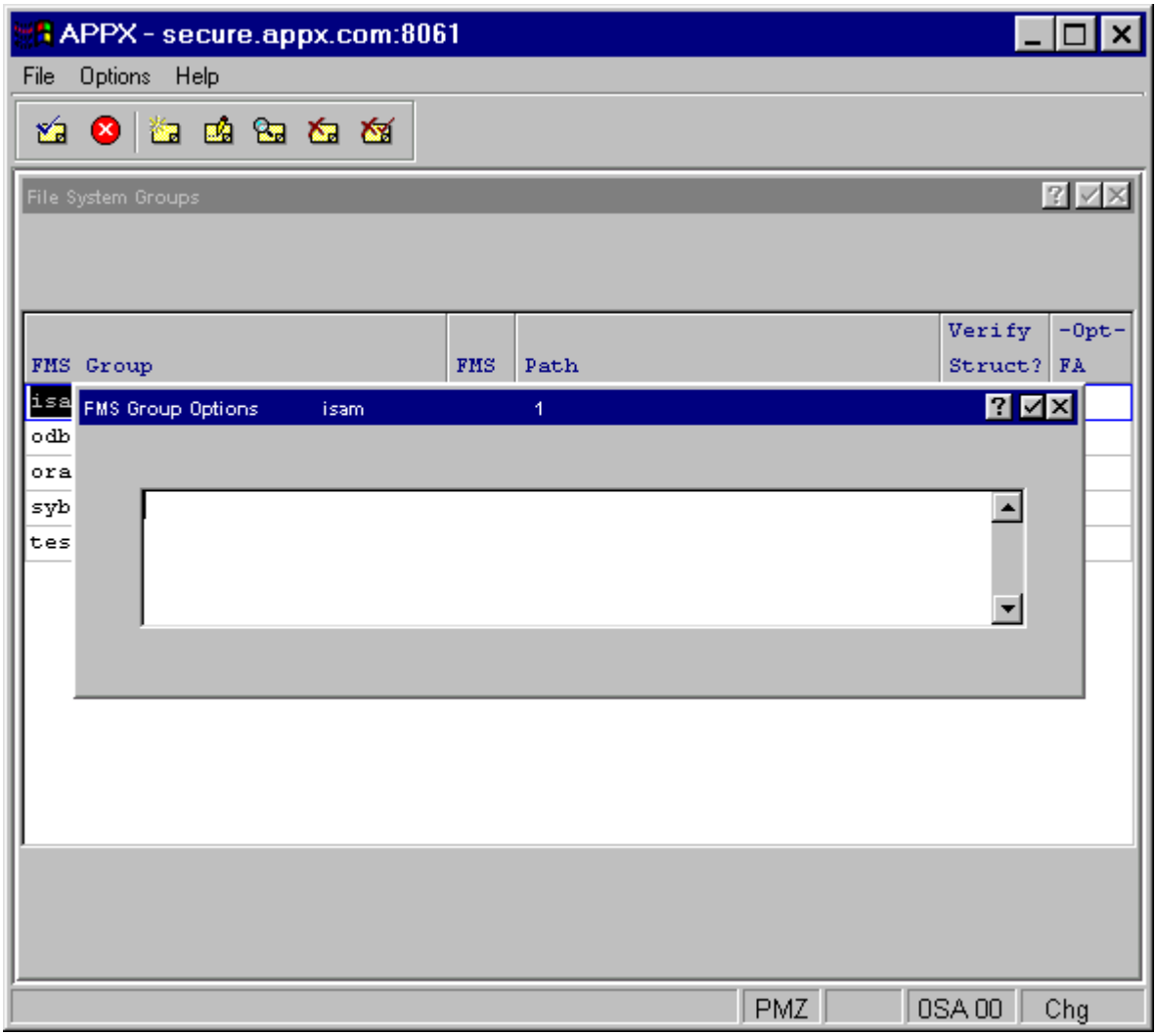


Figure 2-18. FMS Controls Overlay for ISAM (Types 1-3)

FMS Controls Overlay for Sybase

If you select the FMS GROUP ATTRIBUTES option in File System Groups File Maintenance for a file system group with FMS type 4, the FMS Controls overlay for Sybase appears as shown in *Figure 2-19* below.

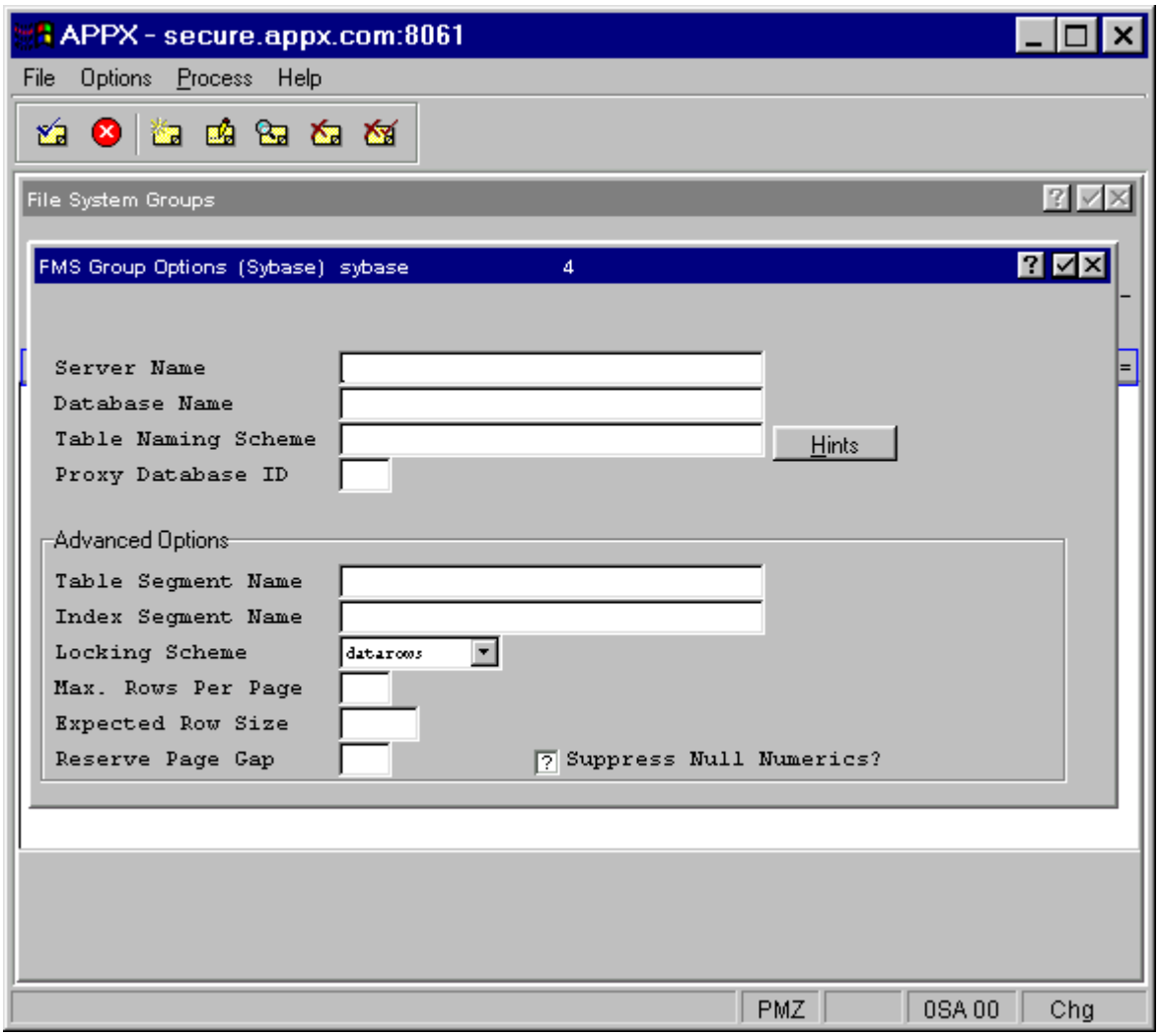


Figure 2-19. FMS Controls Overlay for Sybase (Type 4)

The Sybase overlay contains the fields discussed below. Consult your Sybase database administrator for help in completing these fields.

Server Name

Indicates the name of the server on which the file system resides.

Database Name

Indicates the external name of the database being accessed.

Table Naming Scheme

Assists APPX in translating its 8-character file names to Sybase's external table names. If users will be accessing Sybase at the File System Group level, then a "Table Naming Scheme" is entered here. Select the HINTS option for help on constructing a table-naming scheme. See the Table Naming Scheme Hints section for an example of the Hints overlay. If you intend to override these parameters at the file level in Database Management, then you can leave this field blank and enter an explicit table name there.

Proxy Database ID

Defines the APPX Database ID.

Table Segment Name

Identifies the area within the database to store data for the table.

Index Segment Name

Identifies the area within the database to store index information for the table.

Locking Scheme

Indicates the level of locking for the table:

- *datarows* locks the current record only (the most similar to APPXIO behavior)
- *datapages* locks the entire page or block containing the current record (not recommended)
- *allpages* locks the entire table

Max. Rows Per Page

Defines the blocking factor.

Expected Row Size

Estimates the average size of a row (or record).

Reserve Page Gap

Determines how much space to reserve between pages. This is used to allow room for growth.

Suppress Null Numerics?

Determines whether or not to disable the null numeric feature inherent in the database; answer "yes" ☑ if you want to be consistent with APPXIO behavior which does not have this capability (a zero entry cannot be distinguished from a null value in APPXIO).

FMS Controls Overlay for Oracle

If you select the FMS GROUP ATTRIBUTES option in File System Groups File Maintenance for a file system group with FMS type 5, the FMS Controls overlay for Oracle appears as shown in *Figure 2-20* below.

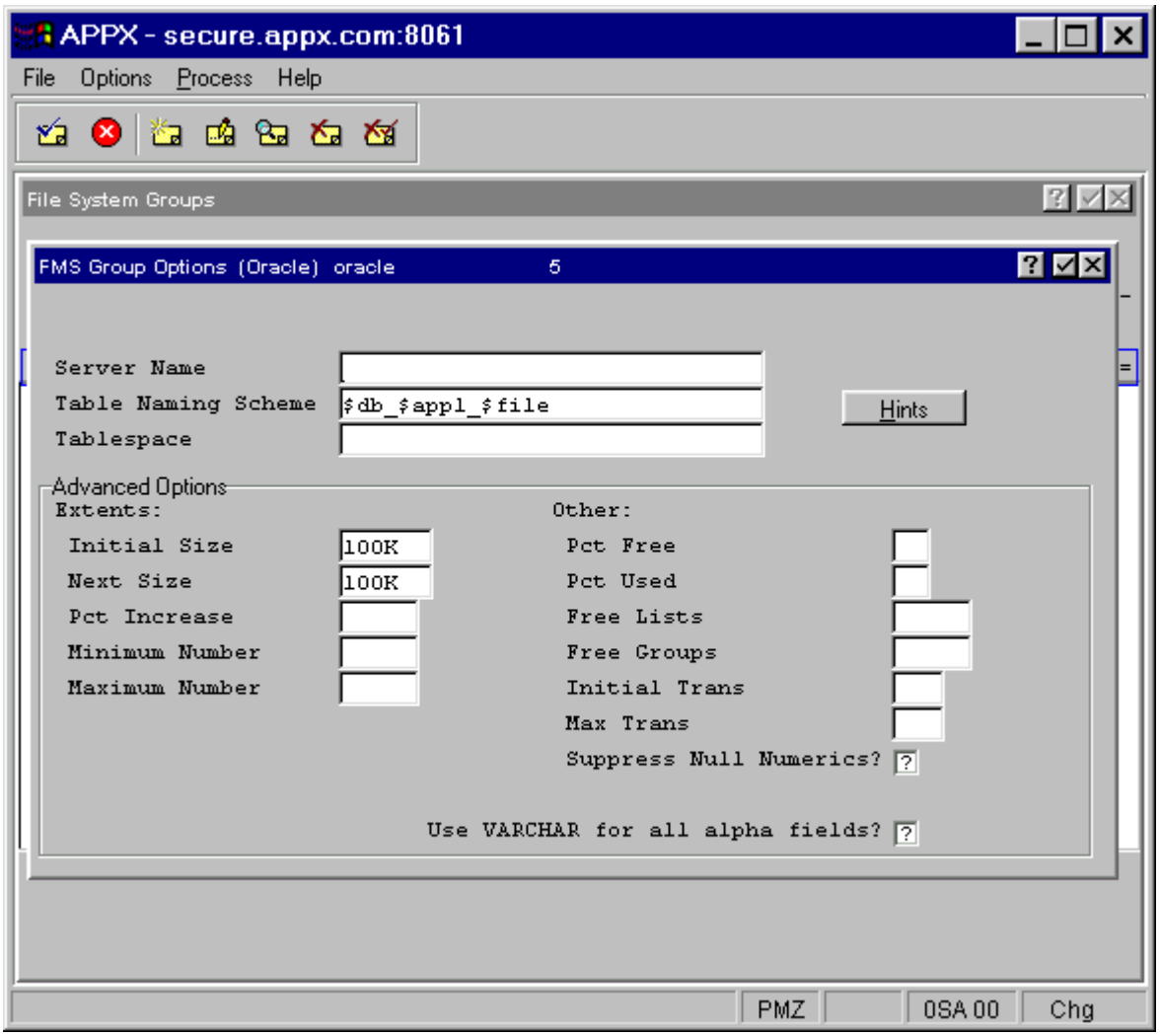


Figure 2-20. FMS Controls Overlay for Oracle (Type 5)

The Oracle overlay contains the fields discussed below. Consult your Oracle database administrator for help in completing these fields.

Server Name

Indicates the name of the server on which the file system resides.

Table Naming Scheme

Assists APPX in translating its 8-character file names to Oracle's external table names. If users will be accessing Oracle at the File System Group level, then a "Table Naming Scheme" is entered here. Select the HINTS option for help on constructing a table-naming scheme. See the Table Naming Scheme Hints section for an example of the Hints overlay. If you intend to override these parameters at the file level in Database Management, then you can leave this field blank and enter an explicit table name there.

Tablespace

Defines the disk allocation for this database. It may be a portion of a file system, or it may include multiple physical disk drives. "Tablespace" is an Oracle term; you should look in their documentation for additional information to help you enter an appropriate value for this field.

Initial Size

Specifies the size of the first extent (block of data stored on disk) in bytes, kilobytes or megabytes. For example, if you want to reserve 100 bytes, enter 100; to reserve 100 kilobytes, enter 100K; to reserve 100 megabytes, enter 100M.

Next Size

Specifies the size of the next extent in bytes, kilobytes or megabytes. As with *Initial Size*, adding "K" or "M" to the number reserves it in kilobytes or megabytes respectively.

Pct Increase

Specifies the percentage by which each extent after the second will increase over the previous extent.

Minimum Number

Specifies the number of extents to be allocated when the database is created.

Maximum Number

Specifies the maximum number of extents to be allocated for this database.

Pct Free

Specifies the percentage of space to be reserved in each data block for future updates to rows contained in that block.

Pct Used

Specifies the minimum percentage of space that will be maintained as used in each data block.

Free Lists

Specifies the number of free lists contained in each free list group of the table.

Free Groups

Specifies the number of groups of free lists for the table.

Initial Trans

Specifies the initial number of transaction entries allocated to each block.

Max Trans

Specifies the maximum number of transaction entries allocated to each block.

Suppress Null Numerics?

Determines whether or not to disable the null numeric feature inherent in the database; answer "yes" ☑ if you want to be consistent with APPXIO behavior which does not have this capability (a zero entry cannot be distinguished from a null value in APPXIO).

Use VARCHAR for All Alpha Fields?

Indicates whether all alphanumeric fields will be defined as variable length, even key fields. Variable length key fields can negatively affect system performance. If this item is set to “yes” ☑, then APPX will assign the *VARCHAR* field type to all alphanumeric fields. If this item is set to “no” ☐, then APPX will assign the *CHAR* field type (fixed length, 30 characters) to key fields and alphanumeric fields of less than six characters, and the *VARCHAR* field type to all other alphanumeric fields.

Table Naming Scheme Hints

If you select the HINTS option from the FMS Controls overlay for Sybase, Oracle, or ODBC in File System Groups File Maintenance, the Table Naming Scheme Hints overlay appears as shown in *Figure 2-20-A* below.

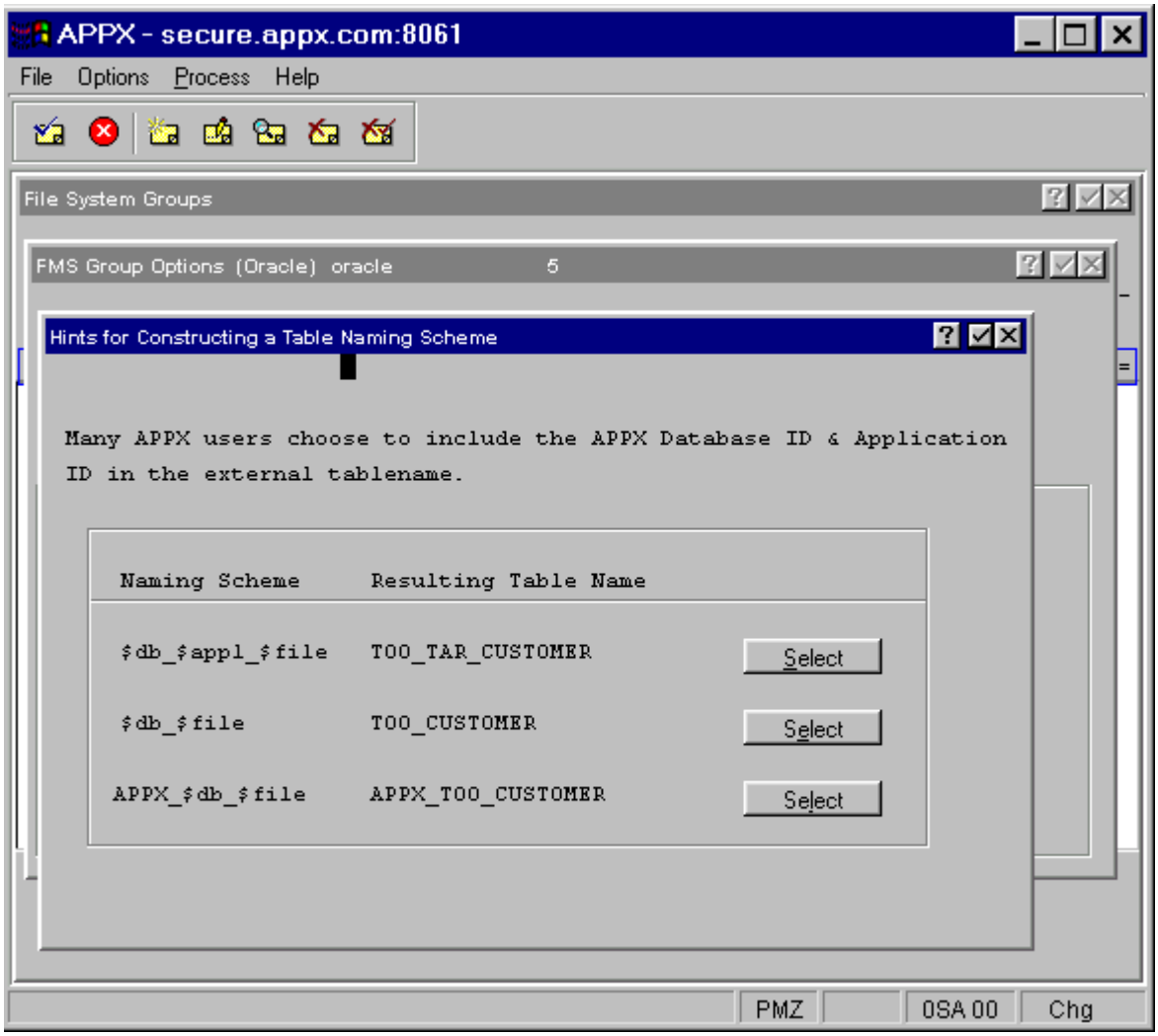


Figure 2-20-A. Table Naming Scheme Hints Overlay

The Hints overlay contains this option by each naming scheme example:

- ***Select*** defaults the associated naming scheme into the FMS Controls overlay.

There are other macros available for use in a table naming scheme besides the examples displayed on the screen. A complete list of macros is provided below:

- $host host name
- $engine APPX engine version
- $endian 'intel' or 'risc'
- $root $APPXPATH
- $uid 3-character APPX User ID
- $user OS User ID
- $pid OS Process ID
- $appl File's application ID

- $ver File's application Version
- $db File's Database ID
- $scope Process nesting level
- $file 8-character file name
- $type "Data", "Struct", "Portdata", etc.
- $anything-else An environment variable

For example, the normal location for an APPX data file would be:

$root/$db/$ap/$type/$file

If you wanted to invert the database/application hierarchy, it would be:

$root/$ap/$db/$type/$file

FMS Controls Overlay for ODBC

If you select the FMS GROUP ATTRIBUTES option in File System Groups File Maintenance for a file system group with FMS type 6, the FMS Controls overlay for ODBC appears as shown in *Figure 2-21* below.

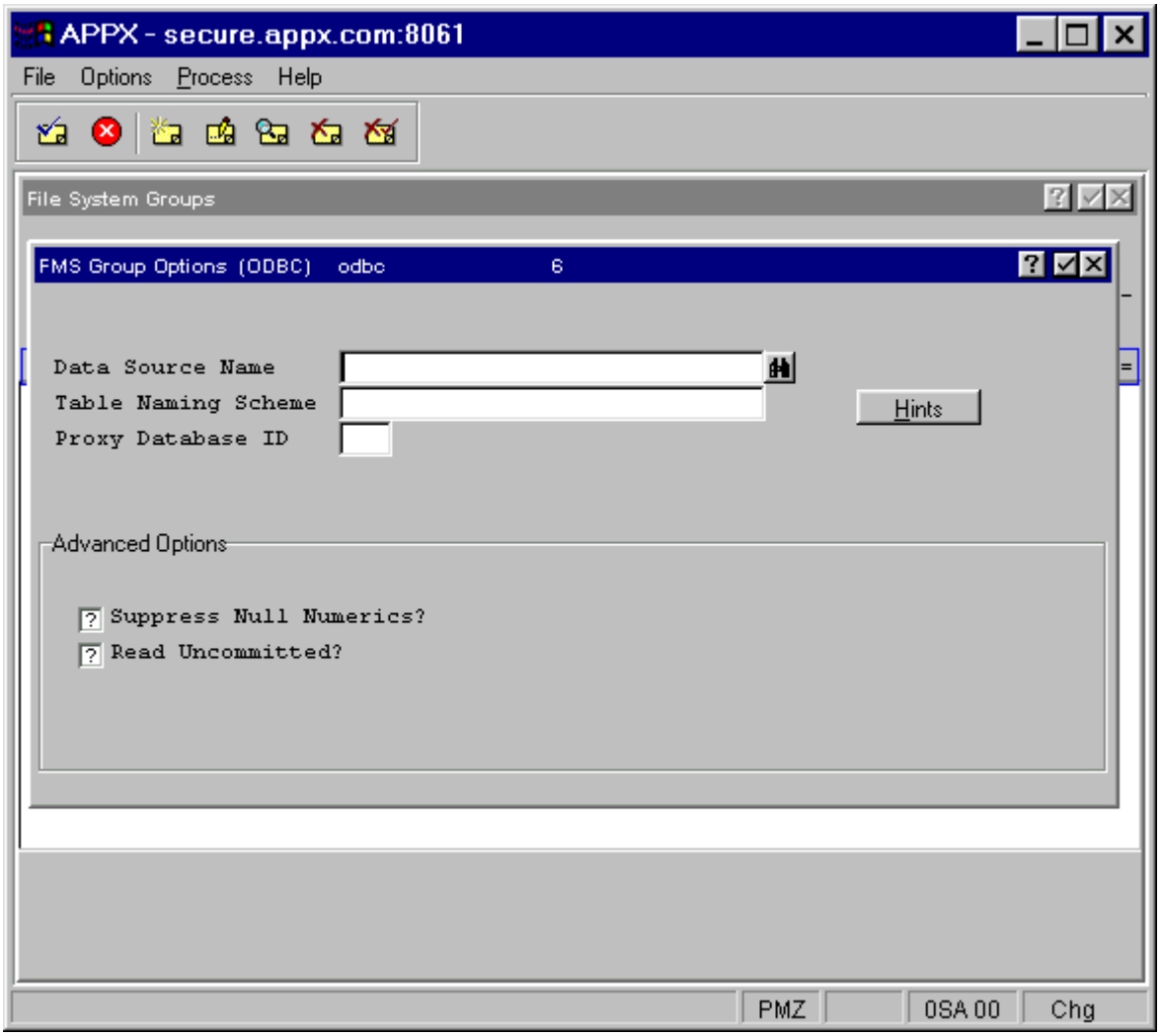


Figure 2-21. FMS Controls Overlay for ODBC (Type 6)

The ODBC overlay contains these fields:

Data Source Name

Defines the particular ODBC source being used. For example, Microsoft Access is an ODBC data source.

Table Naming Scheme

Assists APPX in translating its 8-character file names to the ODBC source's external table names. If users will be accessing the ODBC source at the File System Group level, then a "Table Naming Scheme" is entered here. Select the HINTS option for help on constructing a table-naming scheme. See the Table Naming Scheme Hints section for an example of the Hints overlay. If you intend to override these parameters at the file level in Database Management, then you can leave this field blank and enter an explicit table name there.

Proxy Database ID

Identifies the APPX Database ID.

Suppress Null Numerics?

Determines whether or not to disable the null numeric feature inherent in the database; answer "yes" ☑ if you want to be consistent with APPXIO behavior which does not have this capability (a zero entry cannot be distinguished from a null value in APPXIO).

Read Uncommitted?

Determines whether or not to read a record with an update pending which has not yet been committed; "no" ☐ is recommended in order to emulate APPXIO behavior.

> **Note:** APPX does not support fields (or records) longer than 32,767 (32K) characters. If you map an APPX file to a pre-existing RDBMS table, you may find that the table includes a column larger than 32K; such columns are often referred to as *memo* columns. APPX offers limited support for memo columns. You can access a memo column by defining (in the APPX data dictionary) an ALPHA X(***n***) field in the corresponding position.

When you read a record that includes a memo column, APPX will only read the first ***n*** characters and will truncate any remaining data. If you read a record *with hold* and APPX would have to truncate a memo field, the read will fail. Reading a record *with hold* implies that you may try to update the record and APPX will try to prevent you from losing data through truncation. If you need to delete a record that would include truncated data, use *partial-record I/O* to avoid truncation. You can also use *partial-record I/O* to update other fields in the record. In general, you should avoid reading memo columns except in those cases where you actually need the data.

FMS Controls Overlay for APPX/Net

If you select the FMS GROUP ATTRIBUTES option in File System Groups File Maintenance for a file system group with FMS type 7, the FMS Controls overlay for APPX/Net appears as shown in *Figure 2-22* below.

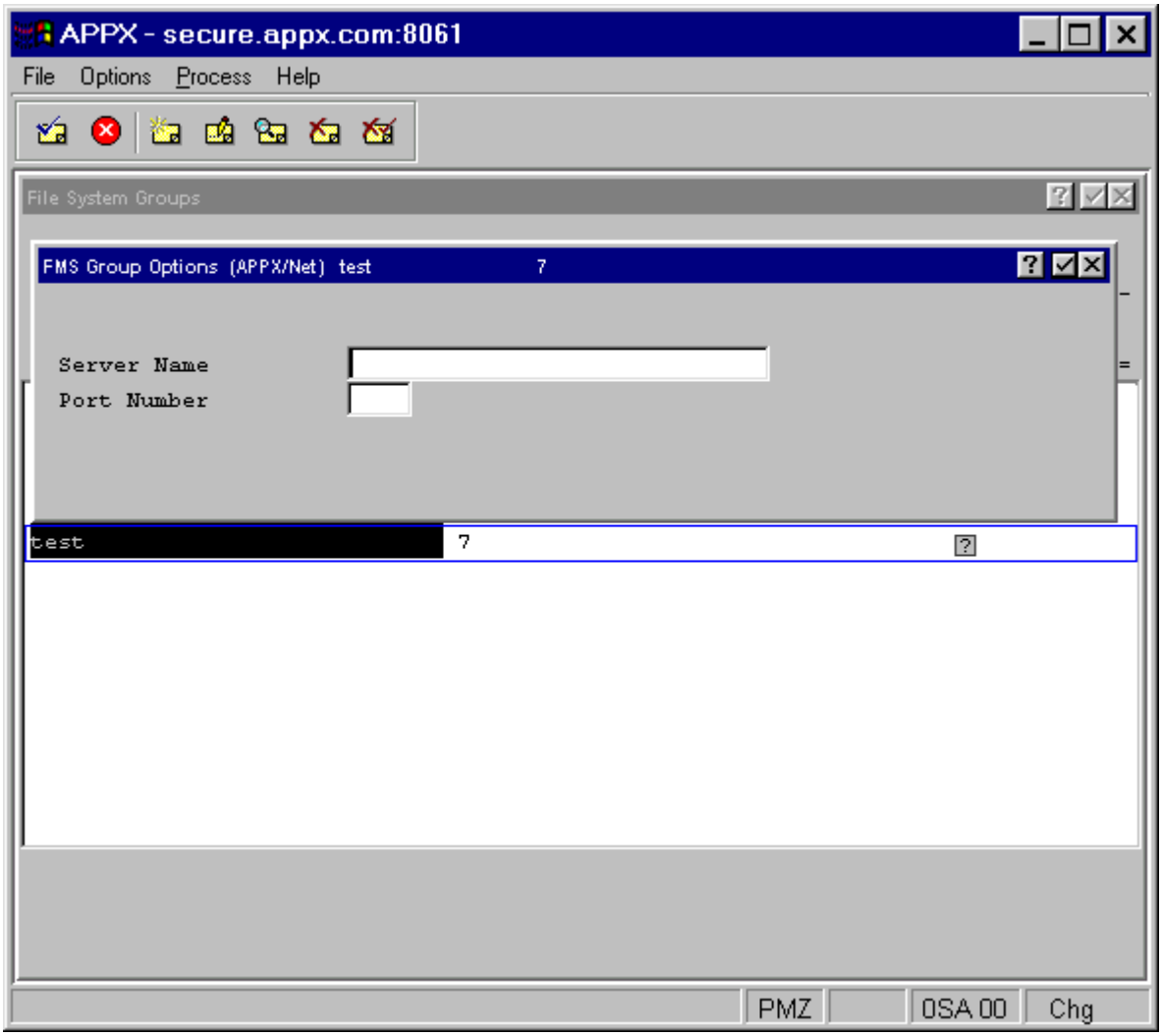


Figure 2-22. FMS Controls Overlay for APPX/Net (Type 7)

Before defining FMS Controls for APPX/Net files, be sure to remove the *FMS Path,* as this field is not used. The APPX/Net overlay contains these fields:

Server Name

Indicates the name of the server on which the file system resides.

Port Number

Corresponds to the port number on which *appxd* (the interface between server and client) is running.

FMS Controls Overlay for DB2

If you select the FMS GROUP ATTRIBUTES option in File System Groups File Maintenance for a file system group with FMS type 8, the FMS Controls overlay for DB2 appears as shown in *Figure 2-22-A* below.

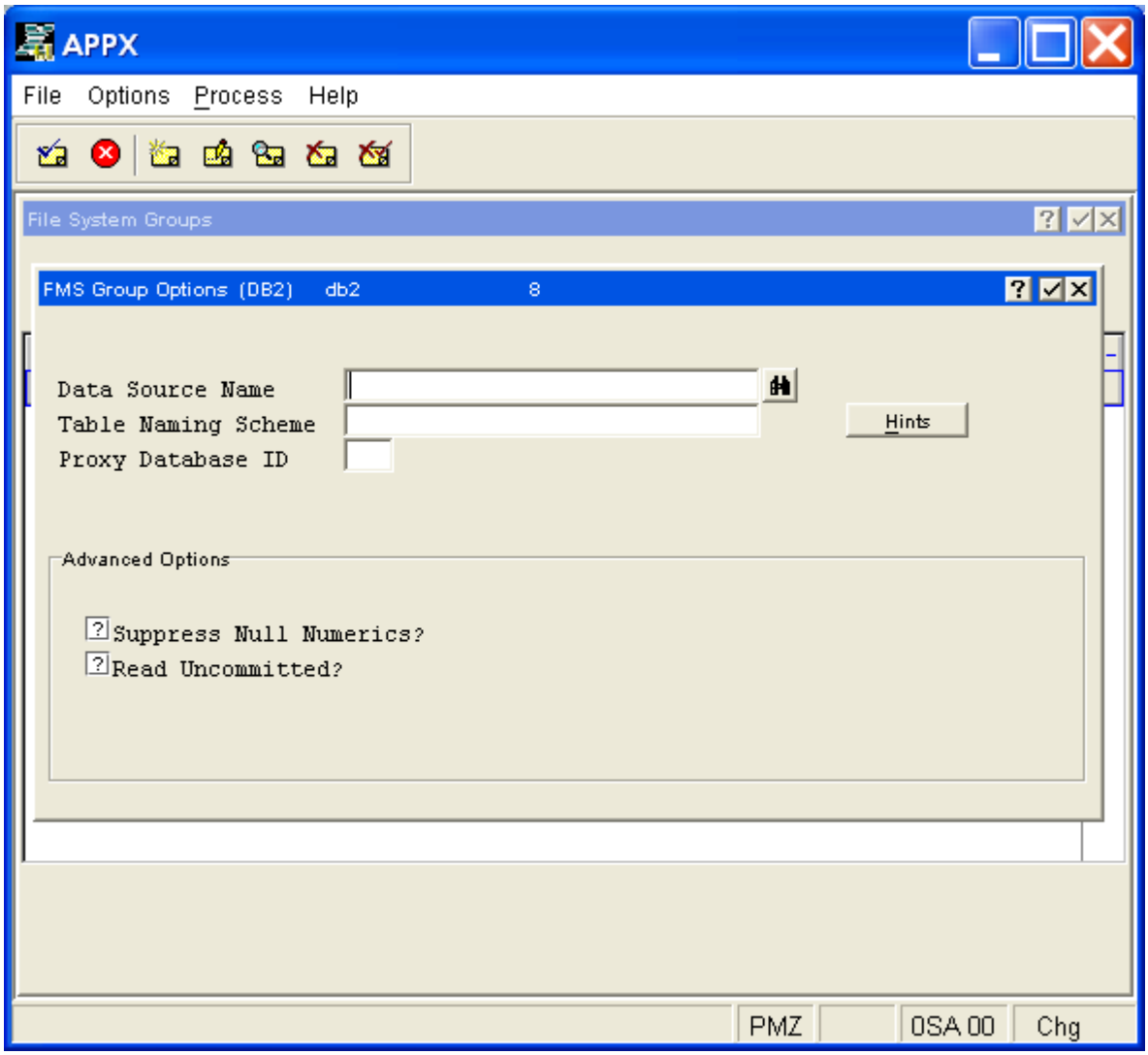


Figure 2-22-A. FMS Controls Overlay for DB2 (Type 8)

The DB2 overlay contains these fields:

Data Source Name

Defines the particular DB2 source being used.

Table Naming Scheme

Assists APPX in translating its 8-character file names to the DB2 external table names. If users will be accessing the DB2 source at the File System Group level, then a "Table Naming Scheme" is entered here. Select the HINTS option for help on constructing a table-naming scheme. See the Table Naming Scheme Hints section for an example of the Hints overlay. If you intend to override these parameters at the file level in Database Management, then you can leave this field blank and enter an explicit table name there.

Proxy Database ID

Identifies the Proxy Database; a proxy database is used for row-level locking. Choose a value that you have not used for a real database.

Suppress Null Numerics?

Determines whether or not to disable the null numeric feature inherent in the database; answer "yes" ☑ if you want to be consistent with APPXIO behavior which does not have this capability (a zero entry cannot be distinguished from a null value in APPXIO).

Read Uncommitted?

Determines whether or not to read a record with an update pending which has not yet been committed; "no" ☐ is recommended in order to emulate APPXIO behavior.

Identities

File System Groups File Maintenance provides an option for IDENTITIES at the bottom of the screen (as does Users File Maintenance which is addressed in the Security section of System Administration). Identity records are maintained by FMS Group/User combination. They provide the place to store login information by FMS Group for each APPX user who requires it, giving the system administrator more flexibility in controlling access.

A “unique” identity is tied to the three character APPX User ID. When the user attempts to connect to an external RDBMS, APPX tries to find an identity record that corresponds to the User ID. If it finds one, the User ID and password designated in the record is passed to the RDBMS. If APPX is unable to find a unique identity record for the User ID, it looks for a “default” identity record.

A “default” identity record for the FMS Group can be set up by creating one with a blank APPX User ID. The User ID and password from this “default” identity is then passed to the RDBMS to gain access. If neither a unique identity nor a default identity exists, then the user’s O/S Login ID and Auxiliary Password (defined in Users File Maintenance) is used. If no User ID and password is found in any of these three areas, the user is unable to access the RDMBS.

The default identity should be used in cases where the user has standard access privileges within the external RDBMS. Unique identities can then be defined for users with more—or fewer—privileges than the “standard” user.

When accessing data in an FMS Group where login information is required (such as Oracle, Sybase or APPX/NET) the login information will be searched as follows:

1. Identity record for FMS Group/User (User ID is present)
2. Default Identity record for the FMS Group (User ID is blank)
3. User’s “O/S Login ID” and “Auxiliary Password”

Note: The use of the user level O/S Login ID and auxiliary password is provided for backward compatibility and may be removed in a future release.

Identities Overlay

The Identities overlay in File System Groups File Maintenance is shown in *Figure 2-23.*

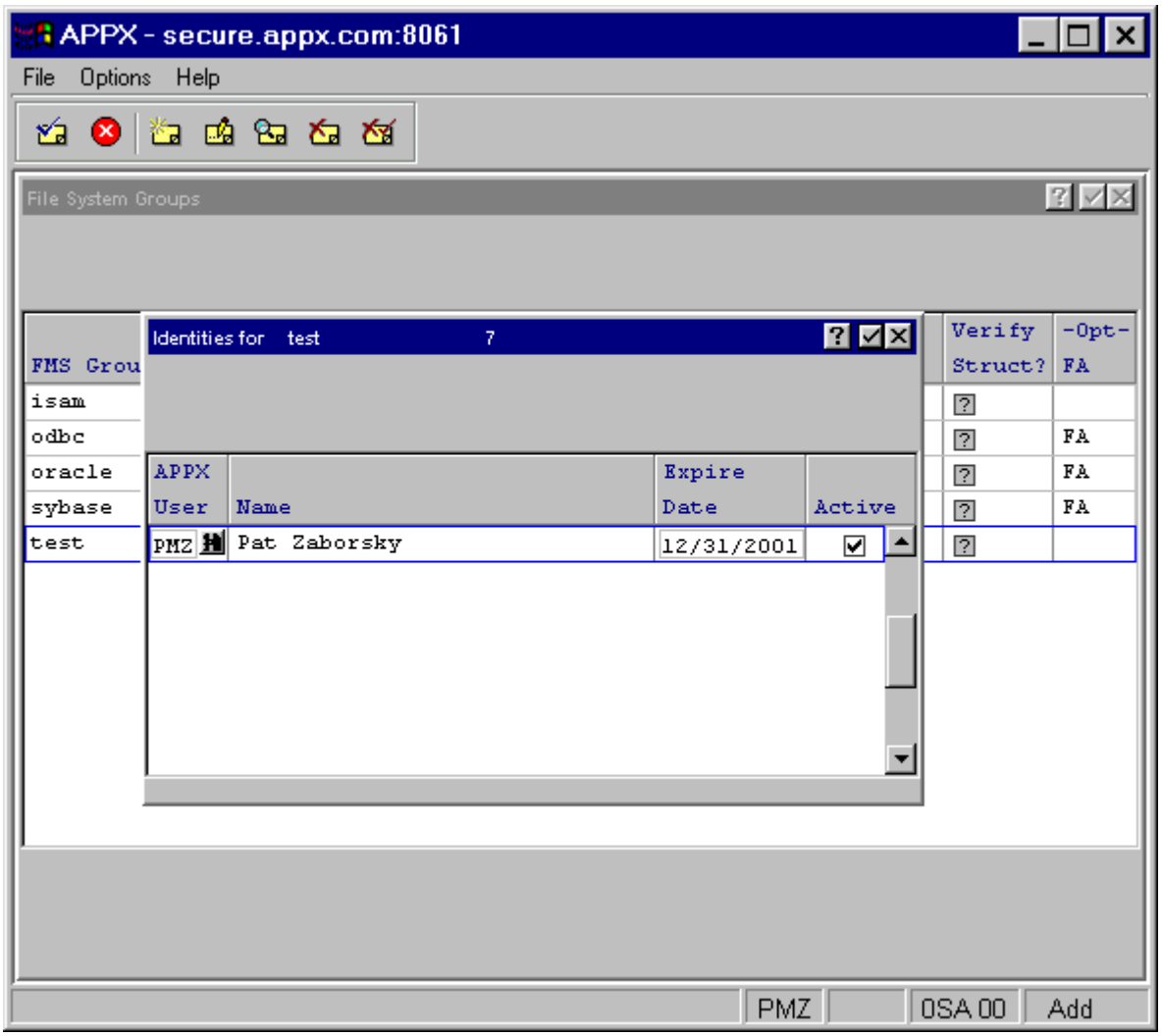


Figure 2-23. Identities Overlay in File System Groups F/M

The Identities overlay contains the following fields:

APPX User

Specifies the user to which this identity is assigned. A “default” identity can be set up for an FMS Group by creating one with a blank APPX User ID.

Expire Date

Specifies the date this identity is no longer valid.

Active?

Can be toggled between “yes” ☑ and “no” ☐ to indicate whether this identity is active or not. This allows you to retain inactive identities that may be needed again in the future.

When you press ENTER from this screen, the Identities Detail overlay described in the next section appears.

Identities Detail Overlay

The Identities Detail overlay accessed from the Identities overlay in File System Groups File Maintenance is shown in *Figure 2-24* below.

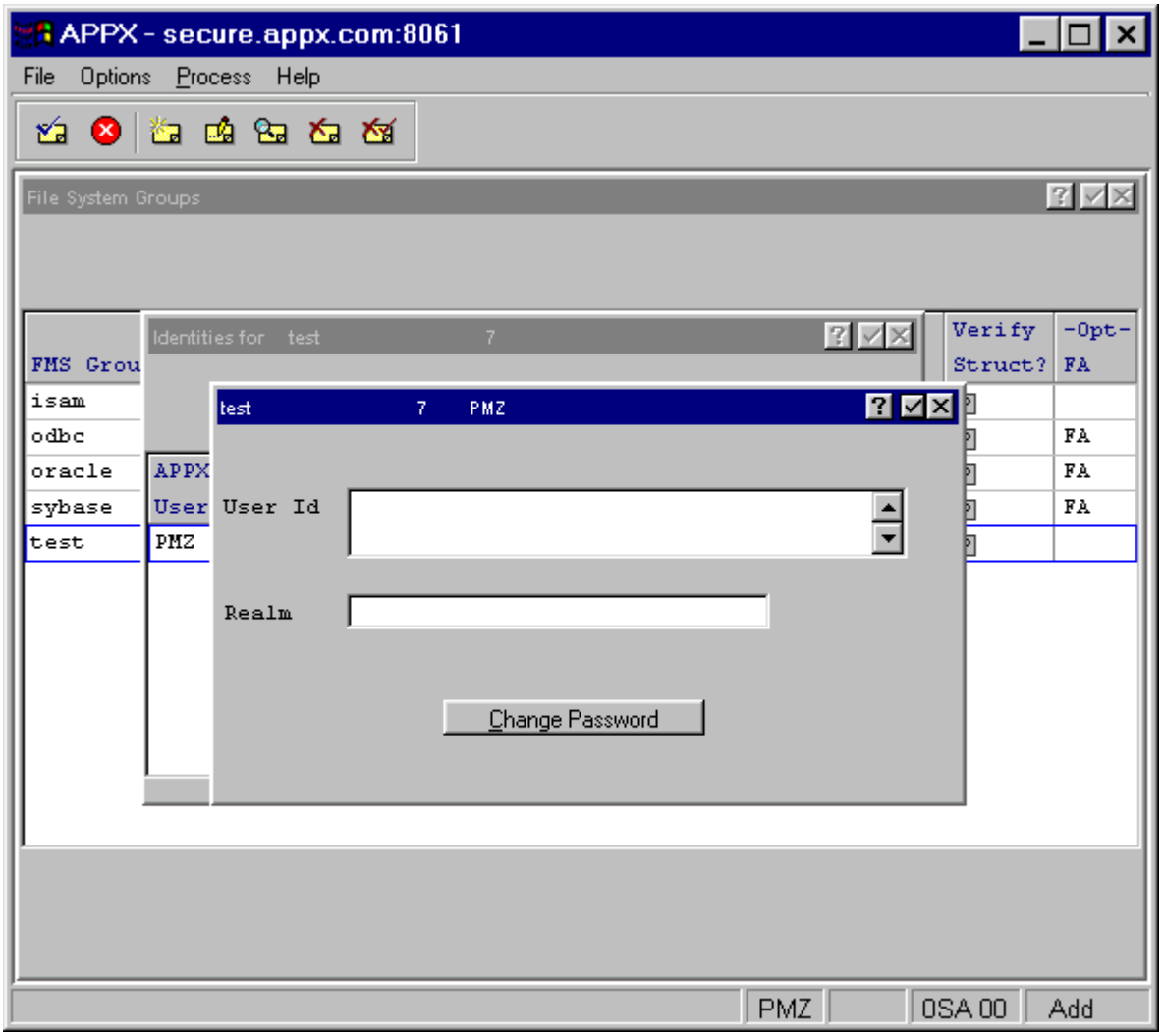


Figure 2-24. Identities Detail Overlay in File System Groups F/M

The Identities Detail overlay consists of the following fields:

User ID

Provides an alternate User ID for this identity when files in this FMS Group are accessed by the specified user. If left blank then the "O/S Login ID" field defined in Users File Maintenance for this user will be used. Likewise, if this identity's password is left blank, the user's "Auxiliary Password" is used.

Realm

Can be used for systems such as NT where more than one authority (domain) may authenticate a user.

The overlay also contains this option:

Change Password

Defines the password to be used in conjunction with the *User ID* when accessing files under this FMS Group.

Disposition Definitions File Maintenance

Disposition Definitions File Maintenance from the Configuration menu is used to define Disposition processes that control the running of jobs and printing of reports. The Disposition Definitions File Maintenance screen is shown in *Figure 2-25:*

Disposition Definitions File Maintenance screen

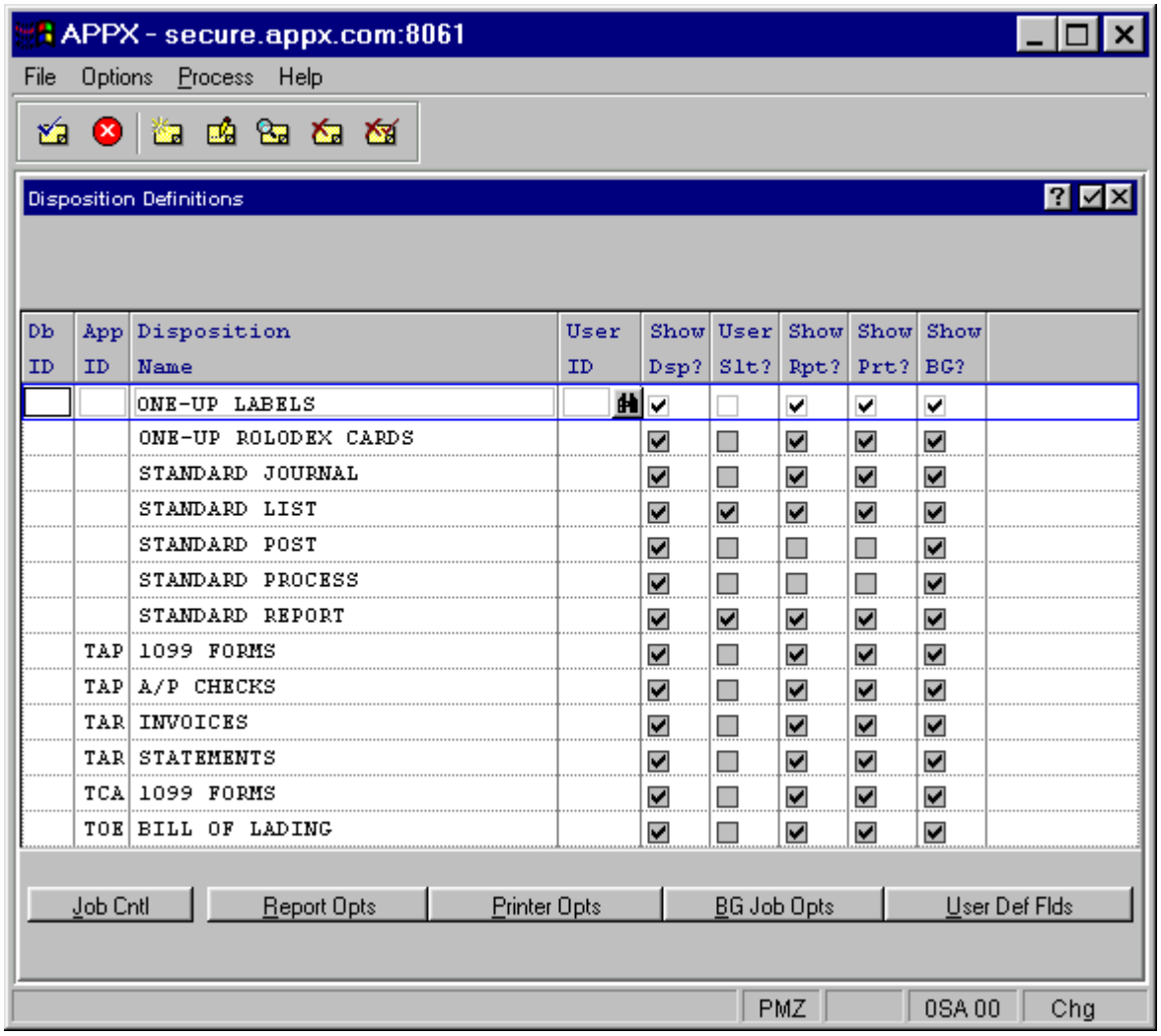


Figure 2-25

The Disposition Definitions File Maintenance screen contains the following fields:

Db ID

Specifies the database ID the disposition is normally used in. If left blank, the disposition can be used in any database.

App ID

Specifies the application ID the disposition is normally used in. If left blank, the disposition can be used in any application.

Disposition Name

Allows you to assign a name to the disposition.

User ID

Specifies the user that normally uses the disposition. If left blank, any user can use the disposition.

Show Dsp?

Indicates whether or not the disposition should be displayed to the user for editing at run time. You may choose not to display the disposition if it contains parameters that should not be modified. Click the checkbox to toggle between "yes" and "no" .

User Sel?

Indicates whether or not a different disposition can be selected by a user at run time to use in lieu of this disposition.

Show Rpt?

Indicates whether or not the Report Options section of the disposition screen should be displayed to the user.

Show Prt?

Indicates whether or not the Print Options section of the disposition screen should be displayed to the user.

Show BG?

Indicates whether or not the Background Job Options screen should be displayed to the user.

The screen also contains these options:

Job Cntl

Displays an overlay that allows you to indicate whether or not the job should be run in background (if available with your operating system). It also allows you to indicate whether or not the user can toggle this checkbox between "yes" and "no" at run time.

Report Opts

Displays an overlay that allows you to define certain report parameters and indicate whether or not they can be edited by users at run time. See the Disposition section of the *APPX User Manual* for descriptions of these parameters.

Printer Opts

Displays an overlay that allows you to define certain printer parameters and indicate whether or not they can be edited by users at run time. See the Disposition section of the *APPX User Manual* for descriptions of these parameters.

BG Job Opts

Displays an overlay that allows you to define certain background job parameters and indicate whether or not they can be edited by users at run time. See the Disposition section of the *APPX User Manual* for descriptions of these parameters.

User Def Flds

Displays an overlay that allows you to create certain user-defined parameters and indicate whether or not they can be edited by users at run time. See the Disposition section of the *APPX User Manual* for descriptions of these parameters.

DATABASES/APPLICATIONS

Databases/Applications Menu

The DATABASES/APPLICATIONS option on the System Administration menu displays the Databases/Applications menu, shown in *Figure 2-26* below. Here you can identify and describe the applications and databases that are to be installed on your computer. When additional applications or databases are developed in the future, you must add them to this set of files as the first step.

Databases/Applications menu

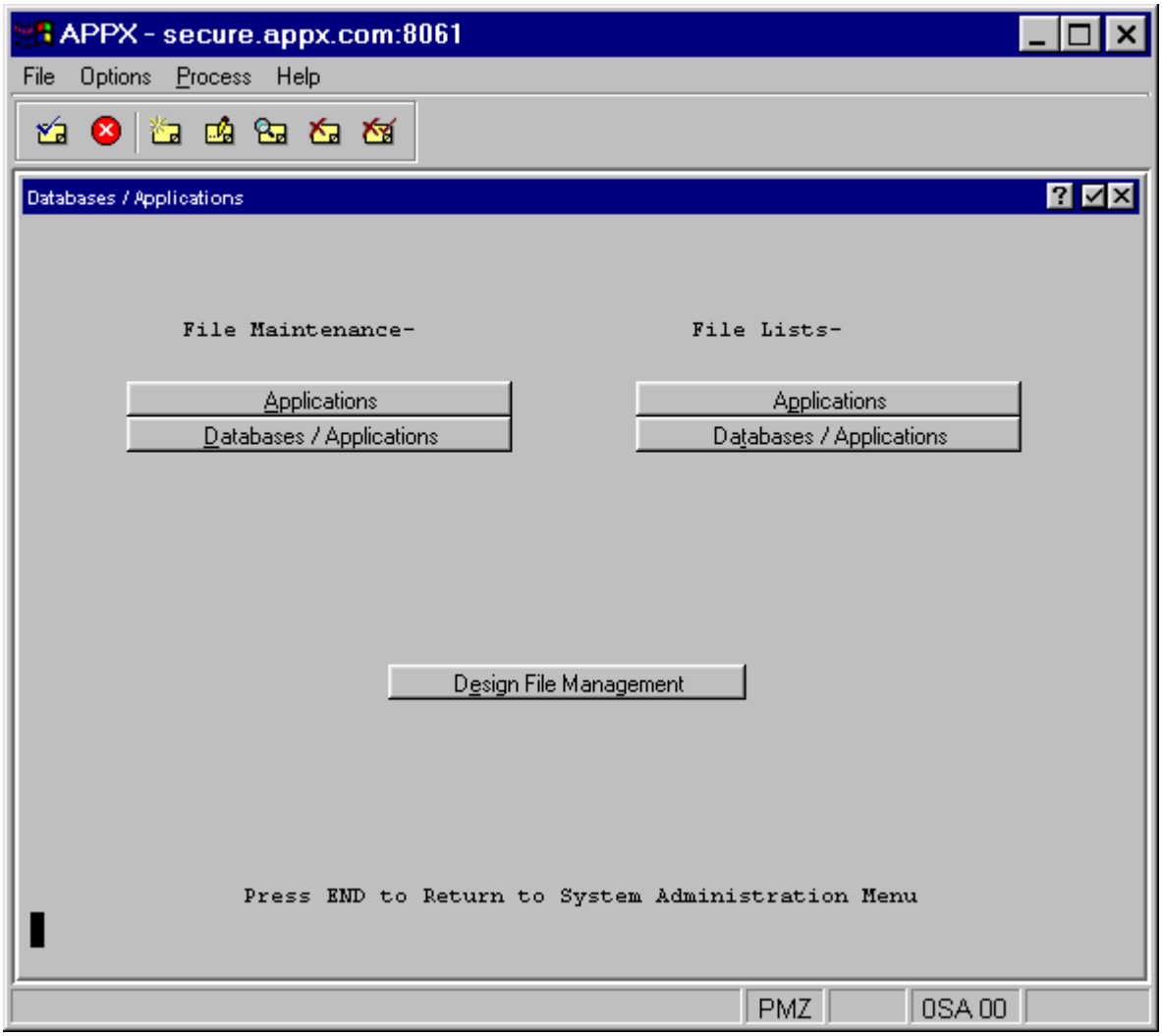


Figure 2-26

File Maintenance

Each of the file maintenance options is described later in this manual. See Applications File Maintenance and Databases/Applications File Maintenance for details.

File Lists

Options are available for printing the complete contents of the Applications and Databases/Applications files.

Design File Management

This option allows you to manage design files for a specified application. Details are provided in the Design File Managment section of this manual.

Applications File Maintenance

You use Applications File Maintenance from the Databases/Applications menu to identify and describe each application that is developed or run on your computer. The Applications File Maintenance screen is shown in *Figure 2-27* below.

Applications File Maintenance screen

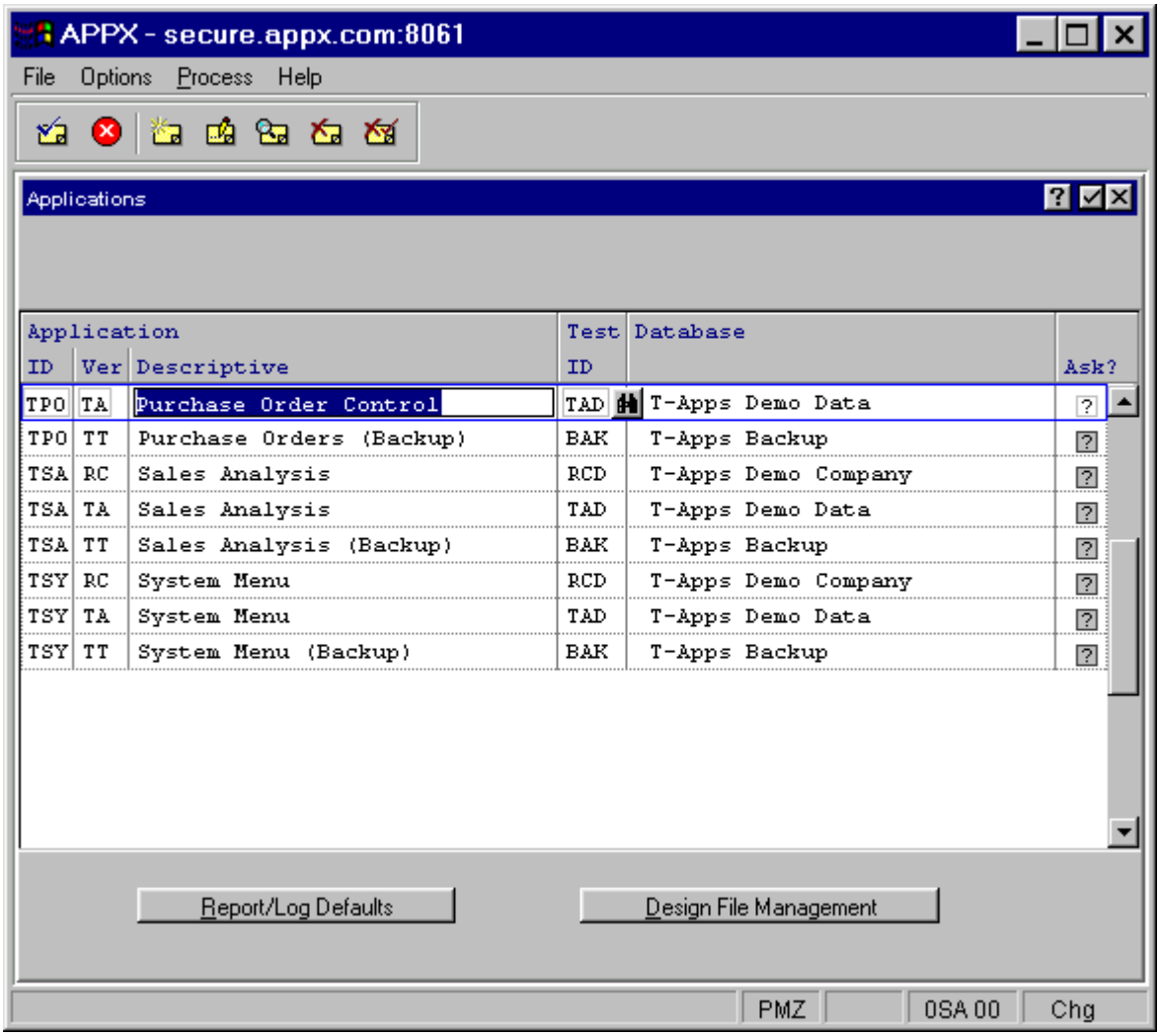


Figure 2-27

The Applications File Maintenance screen contains the following fields:

Application ID/Ver

Specifies the unique identifier of the application, followed by a version number. APPX allows you to have multiple versions of each application so that you may associate customized versions with a single baseline application. Normally, the first (or only) version of each application is *00* (zero, zero).

Descriptive

Is a description for the application/version combination.

Test Database ID

Specifies the identifier that is associated with the test database you want to use while designing this application.

Ask?

Select "yes" ☑ if you want APPX to prompt designers for a database ID when they invoke *Run Test Database* from the Application Design menu, or "no" ☐ if you want APPX to automatically choose the Test Database defined above.

When you press ENTER from this screen, the Applications File Maintenance overlay described in the next section appears.

If you select the REPORT/LOG DEFAULTS option, an overlay allows you to enter customized report/log values for this application. See the Applications Report/Log Defaults section for more information.

If you select the DESIGN FILE MANAGEMENT option, the Design File Management menu appears for the designated application. See the Design File Management section for details.

Applications File Maintenance Overlay

The Applications File Maintenance overlay accessed within Applications File Maintenance is shown in *Figure 2-28* below.

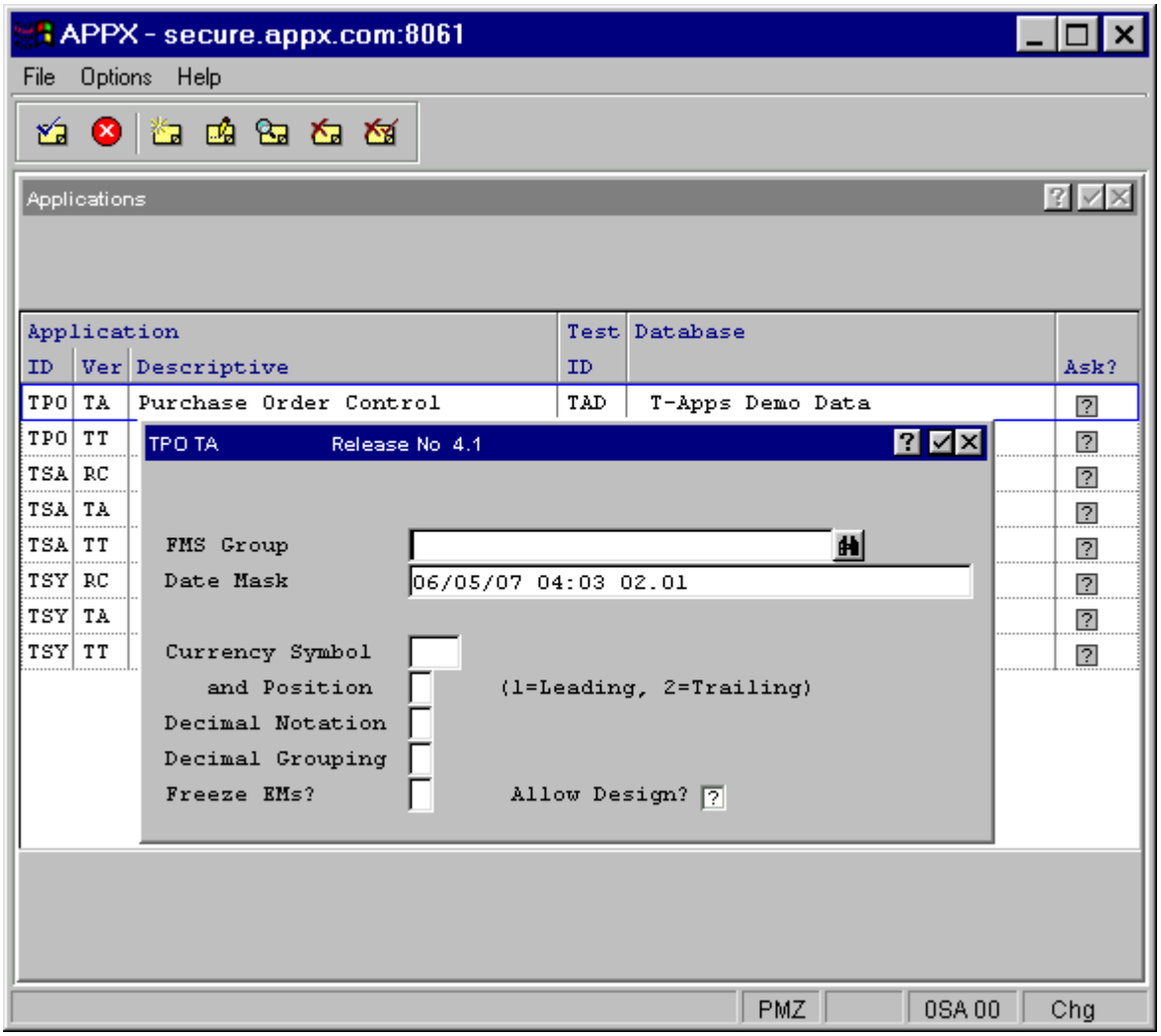


Figure 2-28. Applications File Maintenance Overlay

The fields on this applications overlay are optional. If they are left blank, APPX uses the System Parameters values.

FMS Group

Is the name of the associated file management system group for application design source files.

Date Mask

Specifies the complete mask for how dates and times are to display on screens and reports. One example, a month-day-year-time sequence mask, is: 06/05/07 04:03 02.01.

Currency Symbol

Specifies the currency symbol (such as *$*) that is in effect at your site.

- ***and Position*** specifies whether the currency symbol precedes the numeric values (1) , or follows the numeric values (2).

Decimal Notation

Specifies the one-character symbol that separates the integer and fractional parts of money values. For example, in the United States and Canada, this symbol is a period (.).

Decimal Grouping

Specifies the one-character symbol that separates groups of three digits, representing thousands, millions, etc. In the United States and Canada, this symbol is a comma (,).

Freeze EMs?

Indicates whether or not you want the application's Executable Modules (object code) frozen so that they do not reflect any changes made to the source programs. Click the checkbox to toggle between "yes" ☑ and "no" ☐.

Allow Design?

Indicates whether or not changes can be made to the design files at this time. For example, it might be prudent to "freeze" the design files while upgrading a production system to a new version of an application. Click the checkbox to toggle between "yes" ☑ and "no" ☐.

Press ENTER to save the record.

Applications Report/Log Defaults Overlay

The Applications Report/Log Defaults overlay accessed within Applications File Maintenance and shown in *Figure 2-29* below is the same defaults overlay described previously under System Parameters except that responses here apply only to Application Design. All entries on this screen are optional. If no value is specified, the APPX application design default hierarchy described in the Default Hierarchy section of this manual applies. For field information, refer to the detail descriptions provided under System Parameters.

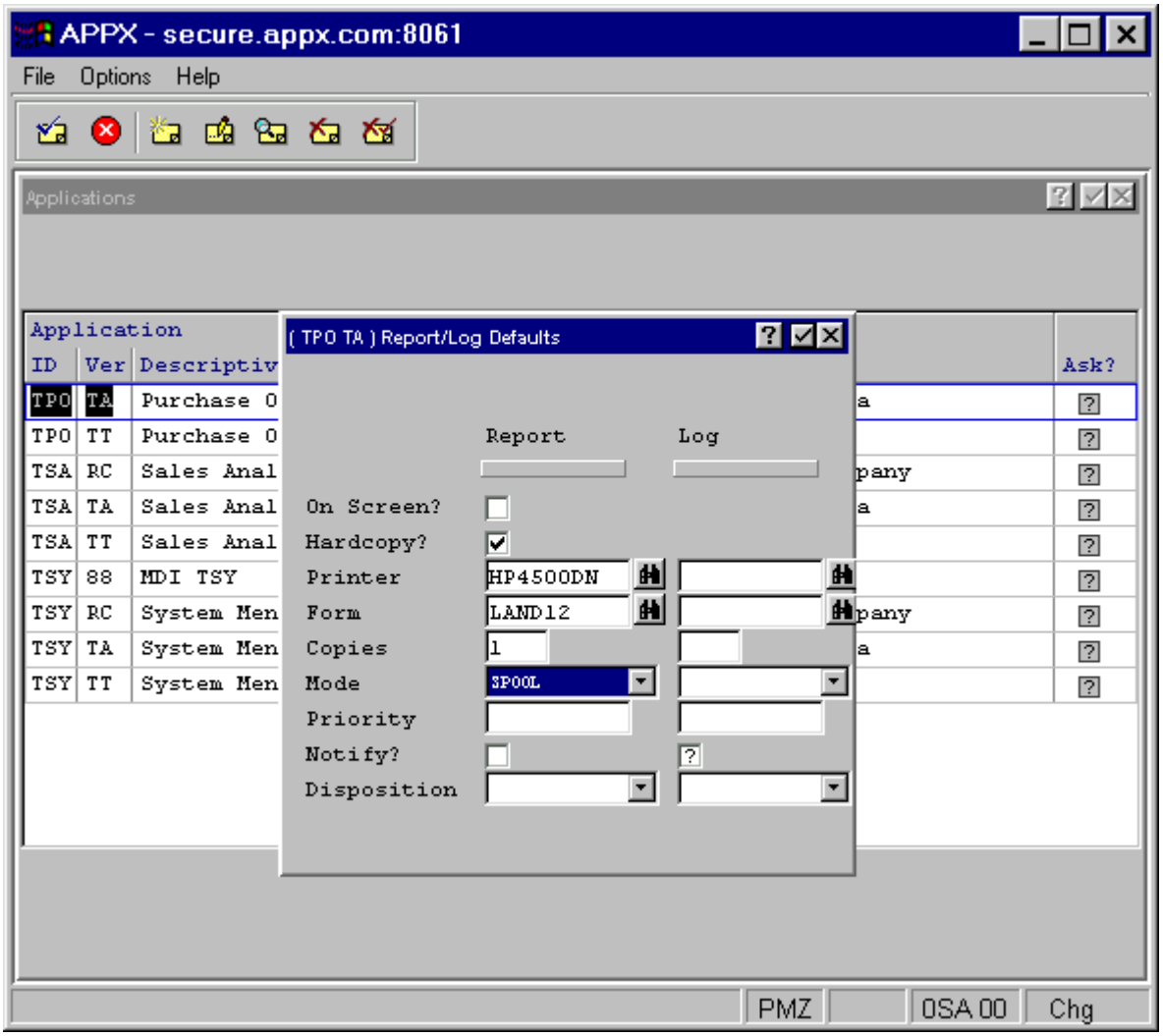


Figure 2-29. Applications Report/Log Defaults Overlay

Databases/Applications File Maintenance

When you select the DATABASES/APPLICATIONS FILE MAINTENANCE option from the Databases/Applications menu, the Databases File Maintenance screen shown in *Figure 2-30* appears. Use it to identify and describe each database that is to be created on your computer.

Databases File Maintenance screen

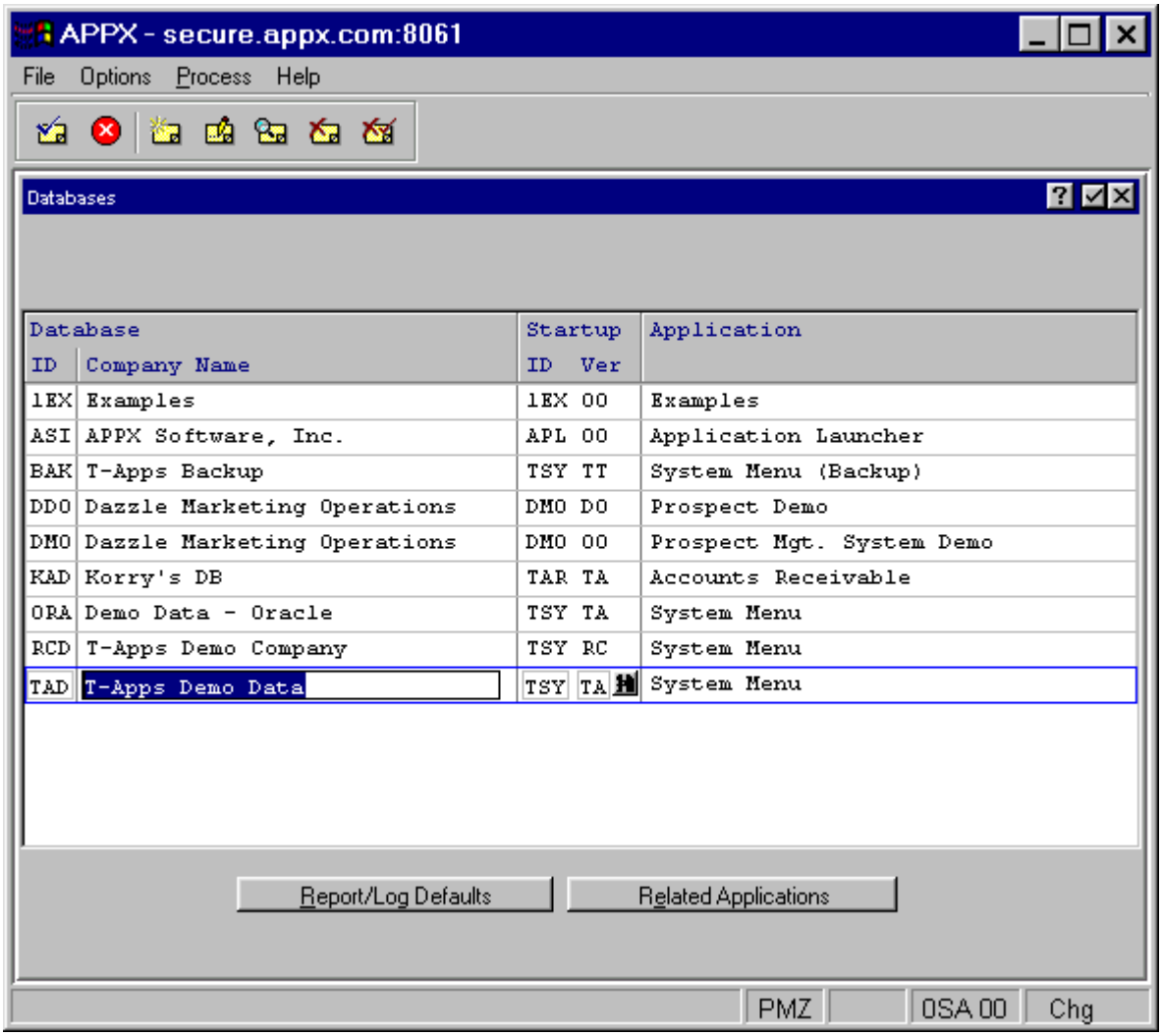


Figure 2-30

The Databases File Maintenance screen contains the following fields:

Database ID

Specifies a unique value for each separate database to be stored in APPX.

Company Name

Is a descriptive name for the database.

Startup Application ID/Ver

Specifies the application and version that will run when a user specifies this database at run time.

When you press ENTER from this screen, the overlay described in the Databases File Maintenance Overlay section appears.

If you select the REPORT/LOG DEFAULTS option, an overlay allows you to enter customized report/log default values for the selected database. See the Databases Report/Log Defaults Overlay section for more information.

If you select the RELATED APPLICATIONS option, an overlay allows you to edit related applications for the selected database. See the Databases Related Applications Overlay section for details. Note that this overlay also appears when you press ENTER from the Databases File Maintenance overlay.

Databases Report/Log Defaults Overlay

If you select the REPORT/LOG DEFAULTS option from the Databases File Maintenance screen, the Databases Report/Log Defaults overlay appears as shown in *Figure 2-31* below. All entries on this screen are optional. If no value is specified, the APPX run time default hierarchy applies. For field details, refer to the System Parameters Report/Log Defaults overlay which is identical to this overlay.

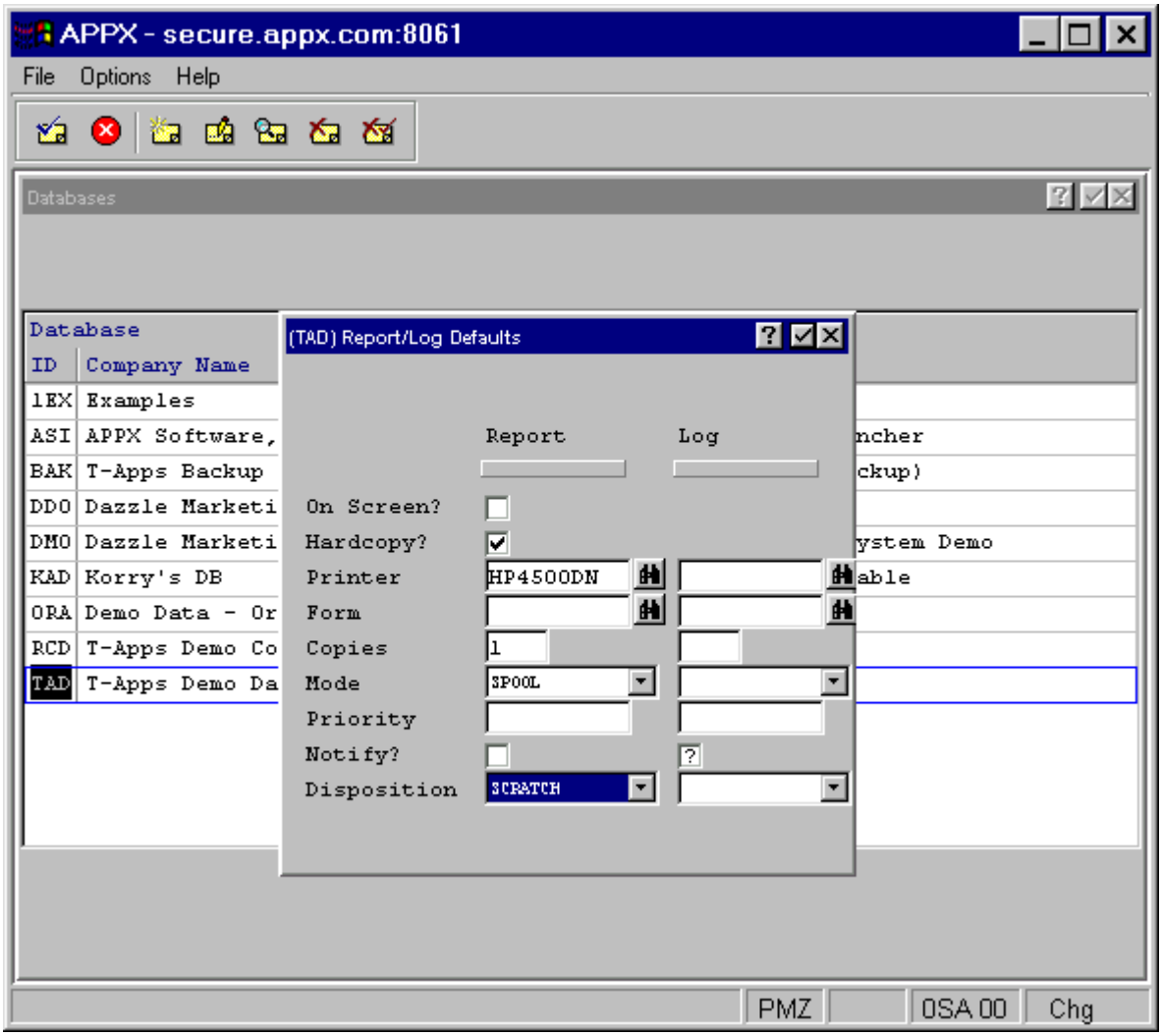


Figure 2-31. Databases Report/Log Defaults Overlay

Databases File Maintenance Overlay

When you press ENTER from the Databases File Maintenance screen, the Databases File Maintenance overlay shown in *Figure 2-32* appears.

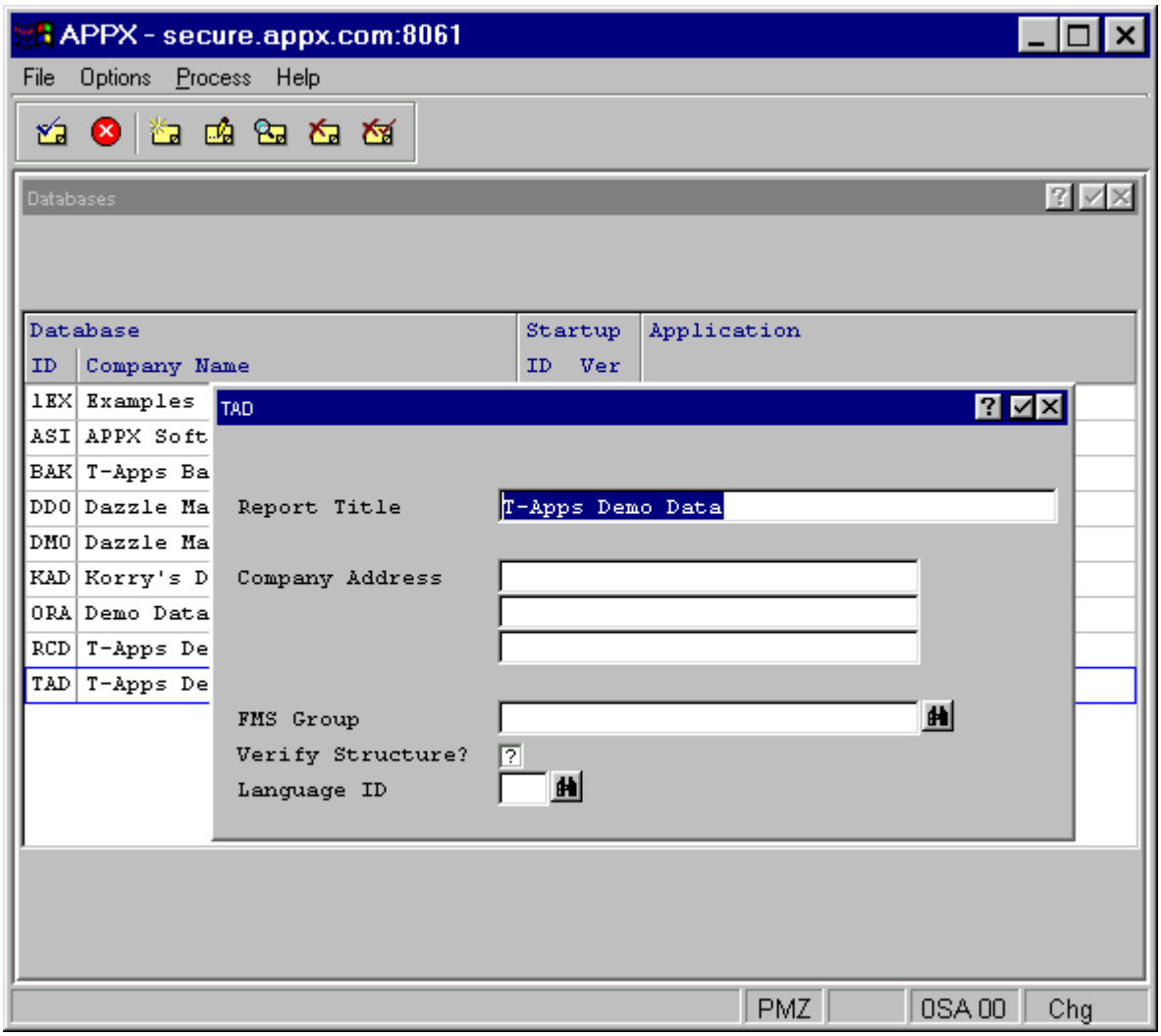


Figure 2-32. Databases File Maintenance Overlay

Some of the fields on this overlay are optional. If they are left blank, APPX uses the *System Parameters* values.

Report Title

Prints on every report that specifies standard headings.

- *Company Address* If this database represents a company, location, or similar entity, enter up to three lines to designate an address that is associated with this database.

FMS Group

Designates one of the previously defined FMS groups as the data path and type of this database. If blank, APPX uses the *System Parameters* value.

Verify Structure?

Controls whether or not the structure of the data files in this database is verified each time a user references it. Select "no" ☐ if you want APPX to skip verification. Select "yes" ☑ if you want the verification to occur. If left unentered ?, APPX uses the *System Parameters* value.

Language ID

Specifies the default presentation language for this database. While a user is accessing data in this database, all applications will be presented in the designated language unless overridden at the application or user level. This field should be left blank if the Language Translation feature is not being used. Language codes can be scanned by clicking the *Scan* icon.

When you press ENTER, the Databases Related Applications overlay described in the next section appears.

Databases Related Applications Overlay

The Databases Related Applications overlay shown in *Figure 2-33* below allows you to enter related applications for the selected database.

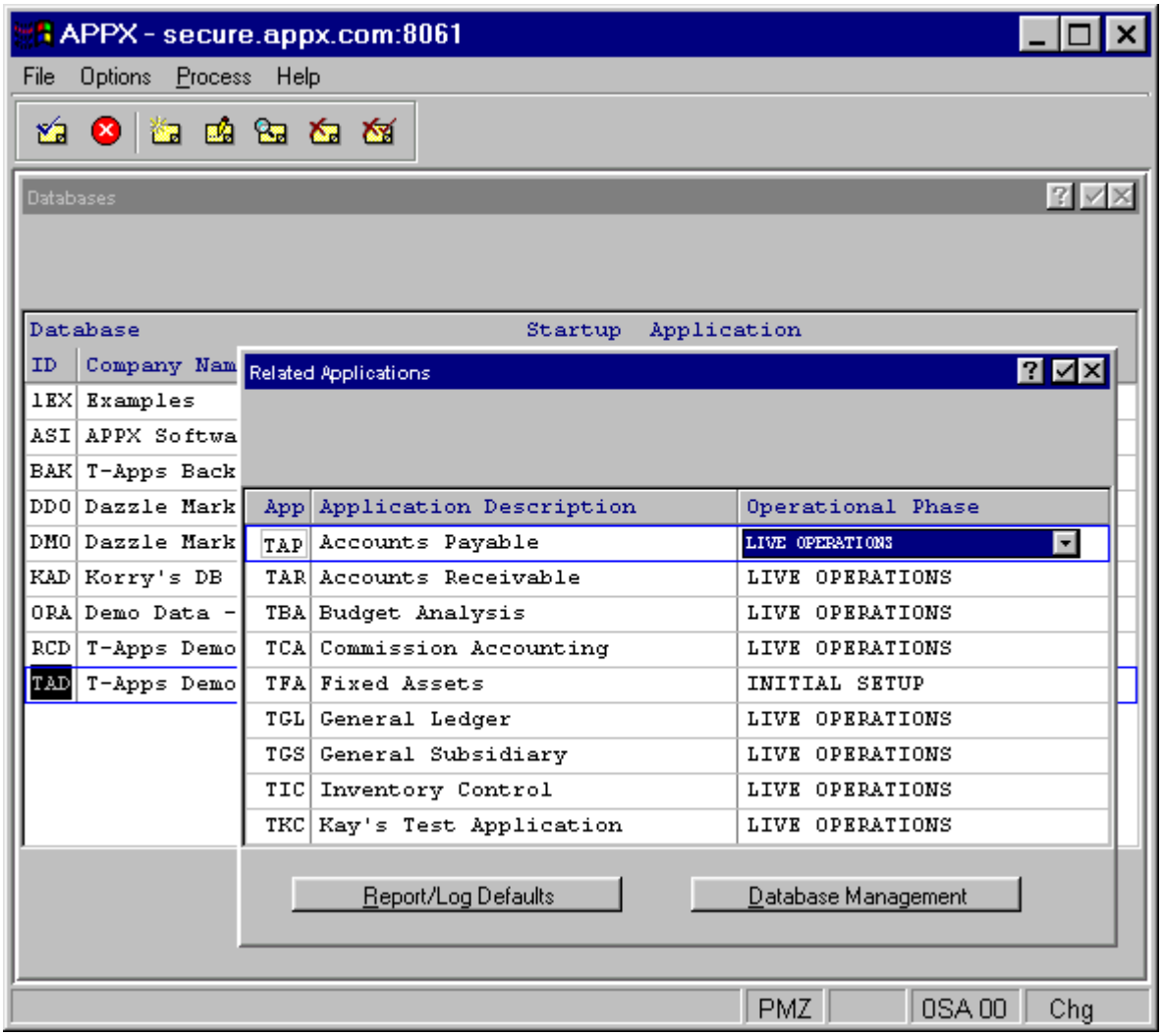


Figure 2-33. Databases Related Applications Overlay

The Databases Related Applications overlay contains the following fields:

App

Defines each of the valid application IDs that are associated with this database. Each one must have been previously specified on the Applications screen. The application description is displayed for the related application.

Operational Phase

Identifies the current operational phase of the database/application. Allowable values are:

- INITIAL SETUP is used during system setup, while you are entering opening data in the files.
- LIVE OPERATIONS is used for everyday, normal processing conditions.
- RECOVERY PROCESSING is used when a problem occurs and you must recover or restore data in files.

When you press ENTER, the Databases Related Applications Language ID overlay described in the next section displays.

If you select the REPORT/LOG DEFAULTS option, an overlay allows you to enter customized report/log default values for the selected database/application. See the Databases/Applications Report/Log Overlay section for more information.

If you select the DATABASE MANAGEMENT option, the Database Management menu appears for the selected database/application. This menu is discussed in detail later in this manual.

Databases/Applications Language ID Overlay

The Language ID specifies the default presentation language for the selected database/application While a user is accessing data in this application for this database, all screens will be presented in the designated language unless overridden at the user level. This field should be left blank if the Language Translation feature is not being used. Language codes can be scanned by clicking the *Scan* icon.

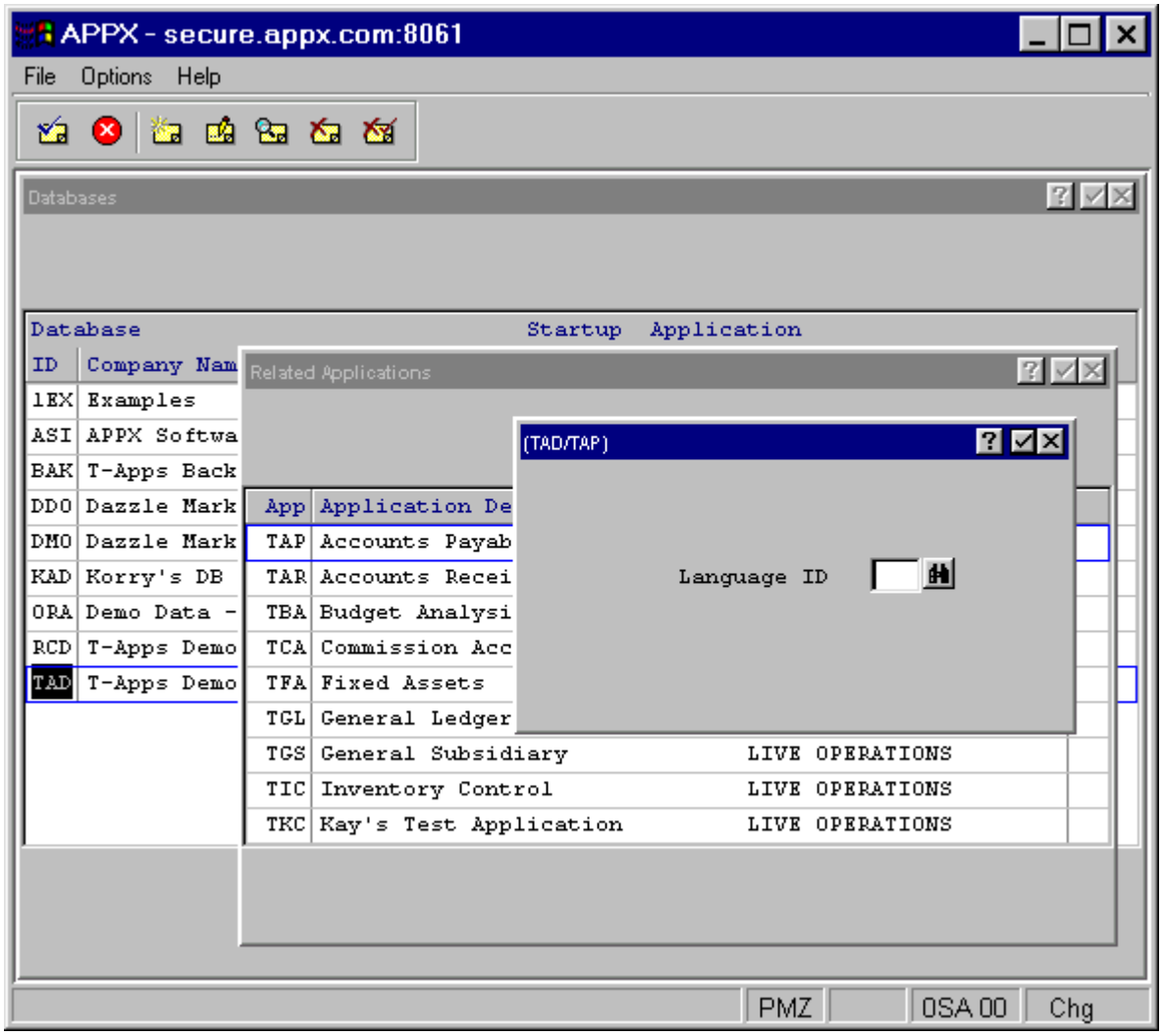


Figure 2-34. Databases/Applications Language ID Overlay

Databases/Applications Report/Log Overlay

If you select the REPORT/LOG DEFAULTS option from the Databases Related Applications overlay, the Databases/Applications Report/Log Defaults overlay appears as shown in *Figure 2-35* below. All entries on this screen are optional. If no value is specified, the APPX run time default hierarchy applies. For field details, refer to the System Parameters Report/Log Defaults overlay which is identical to this overlay.

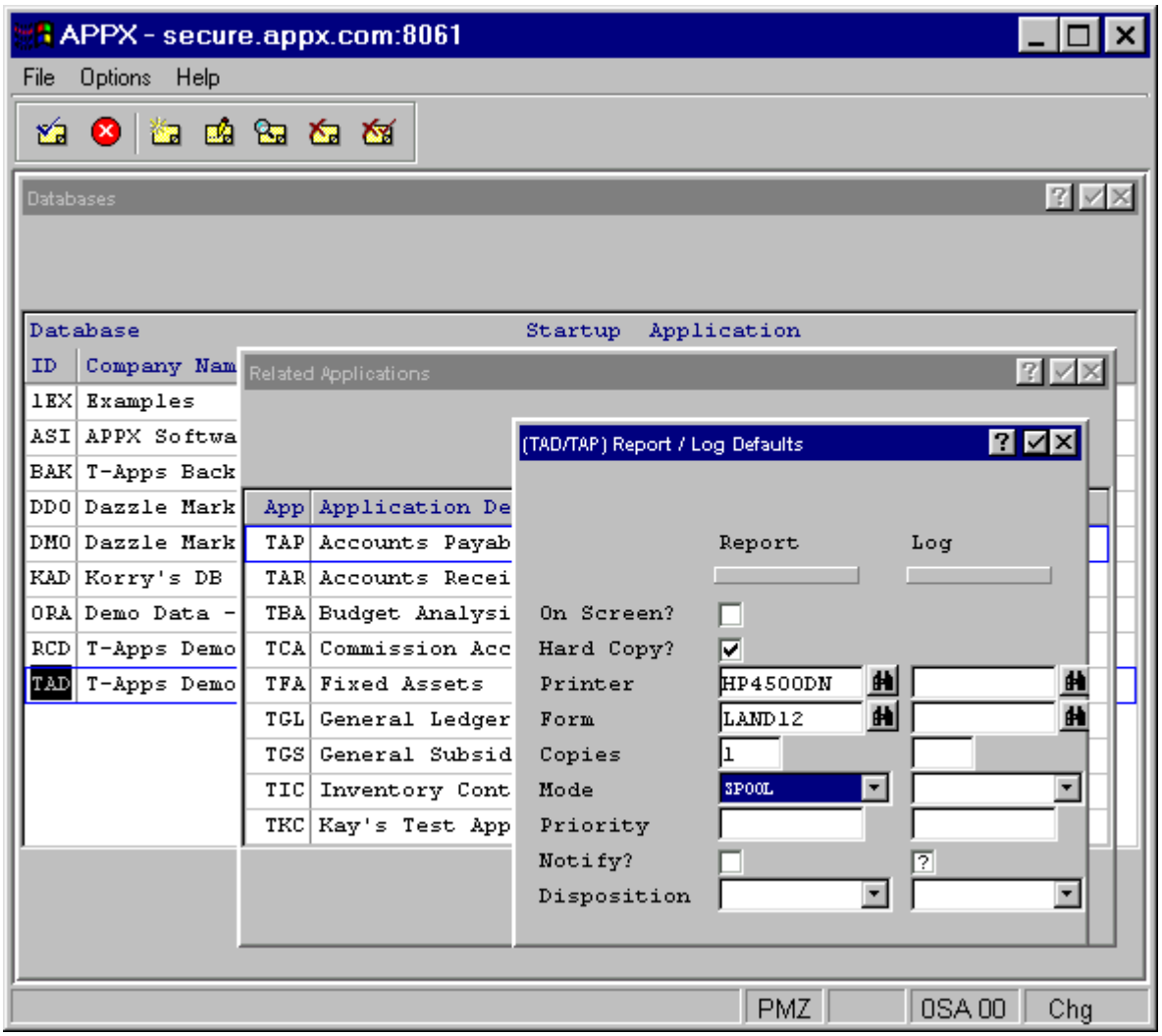


Figure 2-35. Databases/Applications Report/Log Overlay

Design File Management

Design File Management allows you to manage design files for a specified application. When you select the DESIGN FILE MANAGEMENT option from the Databases/Applications menu, the overlay shown in *Figure 2-36* below displays.

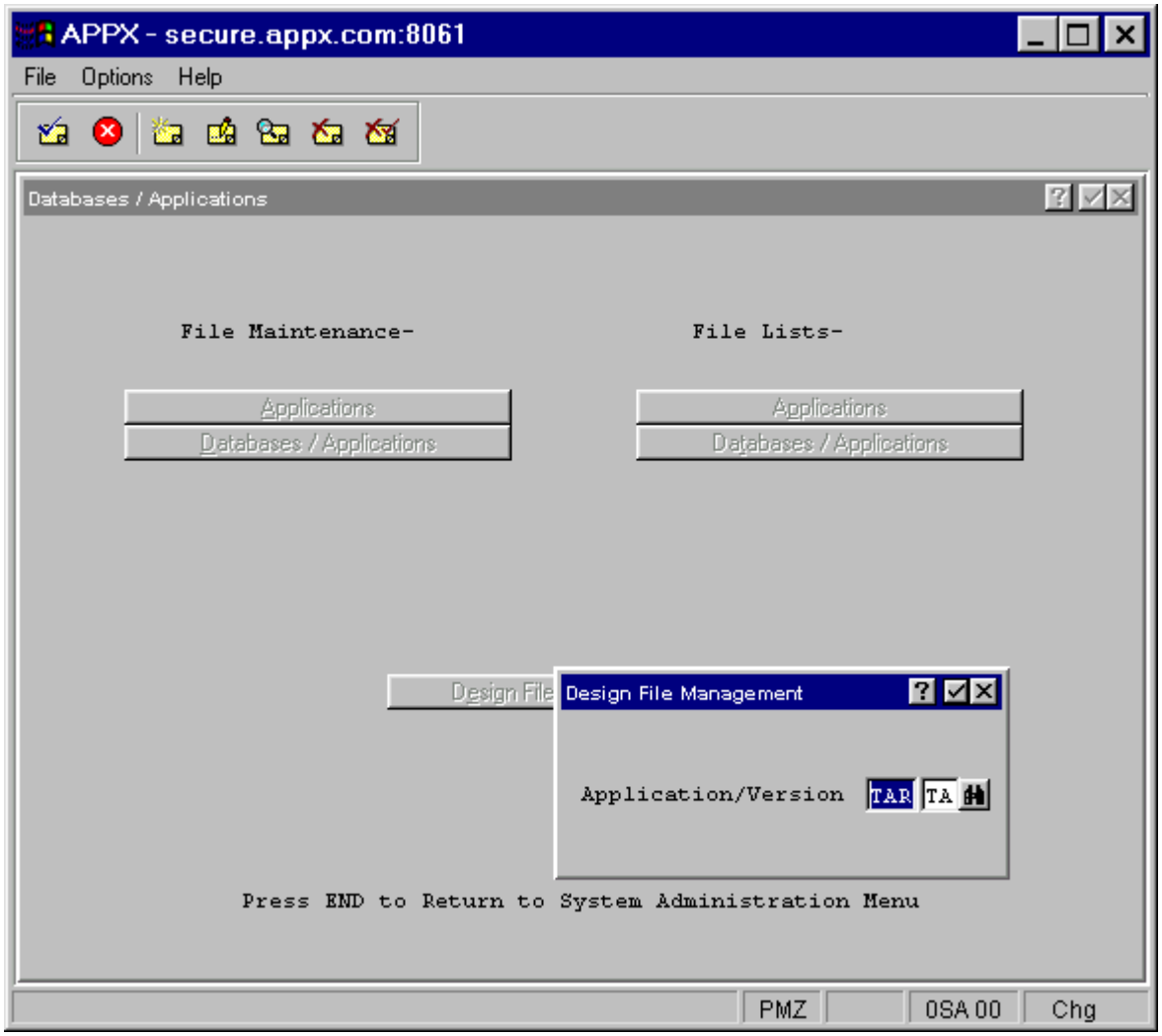


Figure 2-36. Databases/Applications Design File Management Overlay

This overlay requires entry of the following:

- ***Application/Version*** identifies the application ID and version number of the design files you want to manage.

When you press ENTER, the Design File Management menu described in the next section appears.

Design File Management Menu

The Design File Management menu accessed from the Databases/Applications menu is shown in *Figure 2-37* below:

Design File Management menu

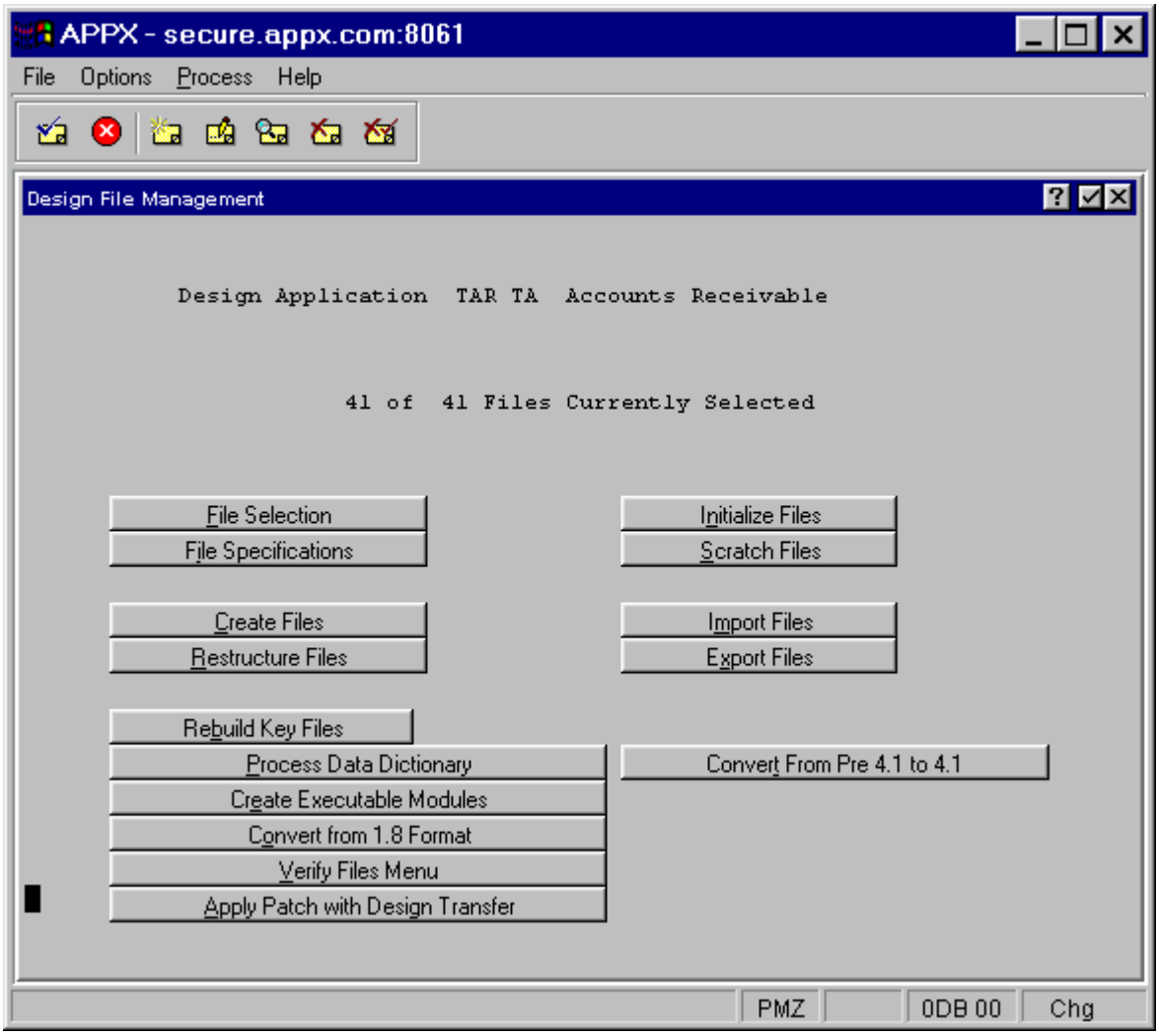


Figure 2-37

The Design File Management menu options are described below.

File Selection

Identifies which files you want to work on. See File Selection in Database Management for more information about the selection process.

File Specifications

Specifies information about the selected files including the estimated number of records in each one. Note that these record counts are not used in actually sizing files (except on the DEC/VAX which is no longer supported). Instead, APPX creates files as small as possible and then allocates disk space dynamically as records are added. However, if you enter a value of *0* (zero), APPX will not create the file when you select the CREATE FILES option. See File Specifications in Database Management for detail field descriptions.

Create Files

Creates files you have defined in the FILE SELECTION option of Design File Management (unless the estimated number of records is zero).

Restructure Files

Rebuilds selected files to reflect structure changes. This includes the addition or removal of fields, or modifications to field lengths, types, or other attributes. For design files, the structures do not normally change from one version to another. However, it is occasionally possible that this option will be necessary. If restructuring is necessary during an upgrade, your upgrade instructions will tell you exactly when and how to do so. If your upgrade instructions do not specifically mention restructuring, do not select this option.

Initialize Files

Deletes the selected files and creates new, empty files. You are prompted to acknowledge that you are deleting data.

Scratch Files

Deletes the selected files. You are prompted to acknowledge that you are deleting data.

Import Files and Export Files

Imports text files into APPX and exports APPX files into text versions, respectively. These options are used to pass files between different types of computers.

Rebuild Key Files

Repairs corrupted index files. APPXIO files consist of two physical files, the "key" file which contains all indexes for a file and the location of their corresponding data in the "data" file. This option deletes the "key" file and reads the "data" file to generate a new "key" file.

This option is not available on Windows/NT systems. You can manually delete the “key” file on these systems, and then run the *Create Files* option to build a new “key” file from existing data.

Process Data Dictionary

Normally, the data dictionary is automatically processed each time you exit a process that changed an element in the data dictionary. However, the *Process Data Dictionary* option allows you to initiate data dictionary processing yourself for select, or all, files and/or work fields. This option is used most often to find and correct errors in the data dictionary.

Create Executable Modules

Compiles processes and creates the executable modules used at run time. This option is used primarily after upgrades or conversions to check for compile errors and make an application run-time ready so that users do not have to wait for executable modules to be created each time they run a process for the first time after modification. All processes, specific process types, or individual processes can be selected.

Convert From 1.8 Format

APPX versions prior to 2.0 used a different file structure. When upgrading from a pre-2.0 version to a 2.0 or later version, this utility must be run to restructure files into the new format as the older structure data files are incompatible with new versions of APPX. Any errors APPX encounters during this process are displayed on your screen and you have the option of printing an error log upon completion.

Verify Files Menu

Allows access to utilities used to verify the structure and integrity of APPX data files. These utilities are described in the Verify Files Menu section.

Apply Patch with Design Transfer

Accesses the design transfer process to transfer all design components from an indicated application into the selected application.

This feature can be utilized by APPX developers to “patch” their applications at customer sites. The developer can create a “patch application” which contains new or modified design components. The patch application can be distributed to customers and transferred into the appropriate application using this feature. The Release Number field in Application Parameters will be set to the value in the patch application.

Convert From Pre 4.1 to 4.1

Converts your design files from the format used before the release of version 4.1 of APPX to the format used in version 4.1. This is necessary because the older design files are no longer compatible with APPX. It actually performs three steps: creates three new files, RESOURCE, RESNAME, and WIDGET; converts character-based boxes to widget boxes; and clears a field in the Processes Header that can now contain Partial Record I/O information.

Verify Files Menu

The Verify Files menu accessed from Design File Management is shown in *Figure 2-38* below:

Verify Files menu

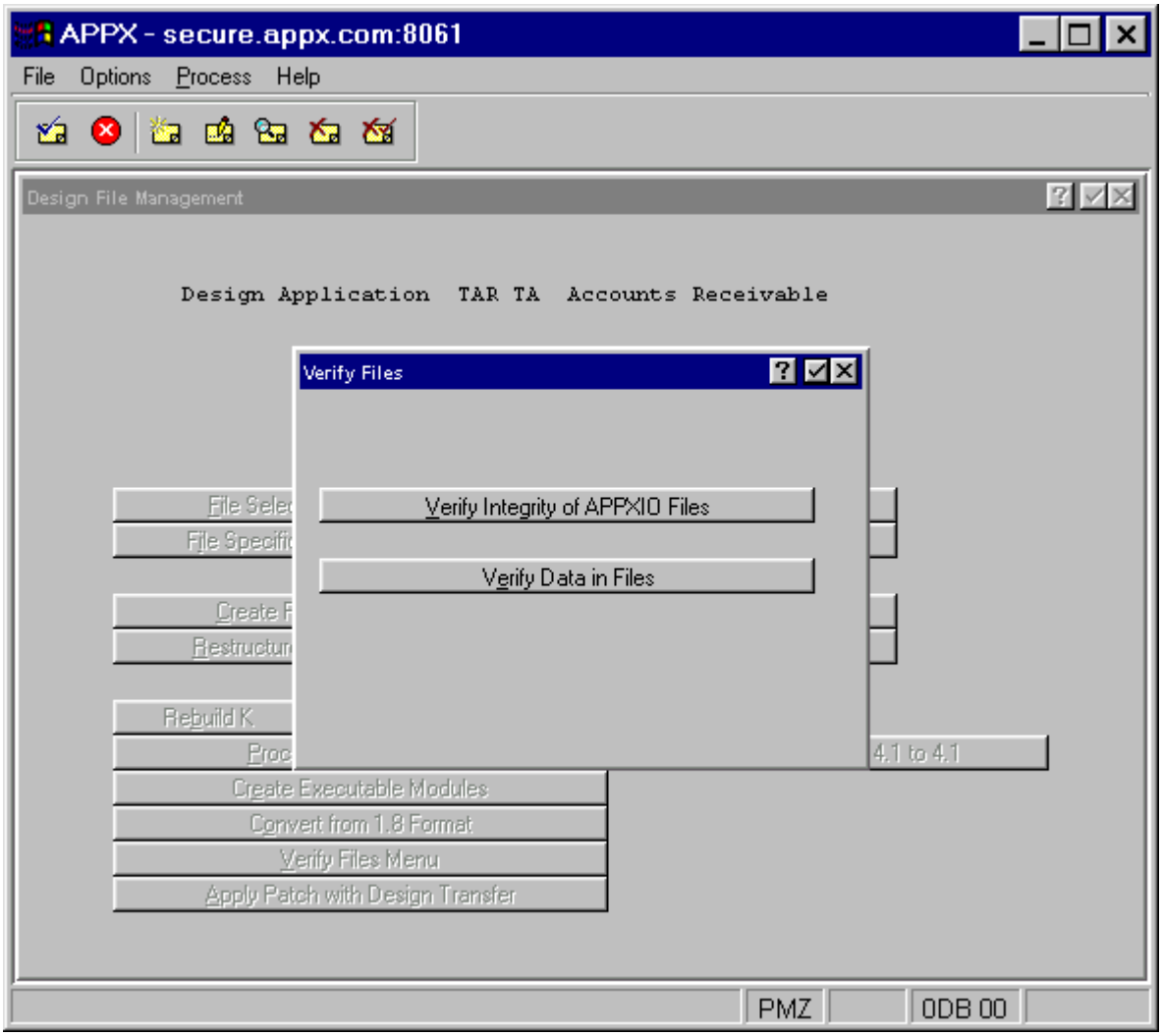


Figure 2-38

The utilities available on this menu are:

Verify Integrity of APPXIO Files

By checking that the entire data file (extension of .dat) is readable and that each record in the file contains a matching record in the index file (extension of .key). If the file passes the integrity check, the file's record count and pathname are displayed. Files that fail the check can be further analyzed by running *appxI* at the command line with a switch of *"-an"*. As the title implies, this option only works on APPXIO-format files (FMS type 1).

Verify Data in Files

Checks each record in the file to ensure that the contents in each field meets the criteria of the field's field type. This option will work on any type of data file. Optionally, referential integrity can be verified by checking references to data lookups to ensure their files and keys exist. For each file with errors, an error log is created in your default print directory.

When you run the *Verify Data in Files* utility, the overlay shown in *Figure 2-39* appears offering these options:

- Include Referential Integrity Check? causes each field with a "Data Lookup" defined to be verified against the parent data lookup file. Records with field values not in the parent file will be reported.
- Report "Null" Data Lookup Values? If referential integrity check is selected, then this flag indicates whether null (blank) values will be validated against the parent data lookup file.

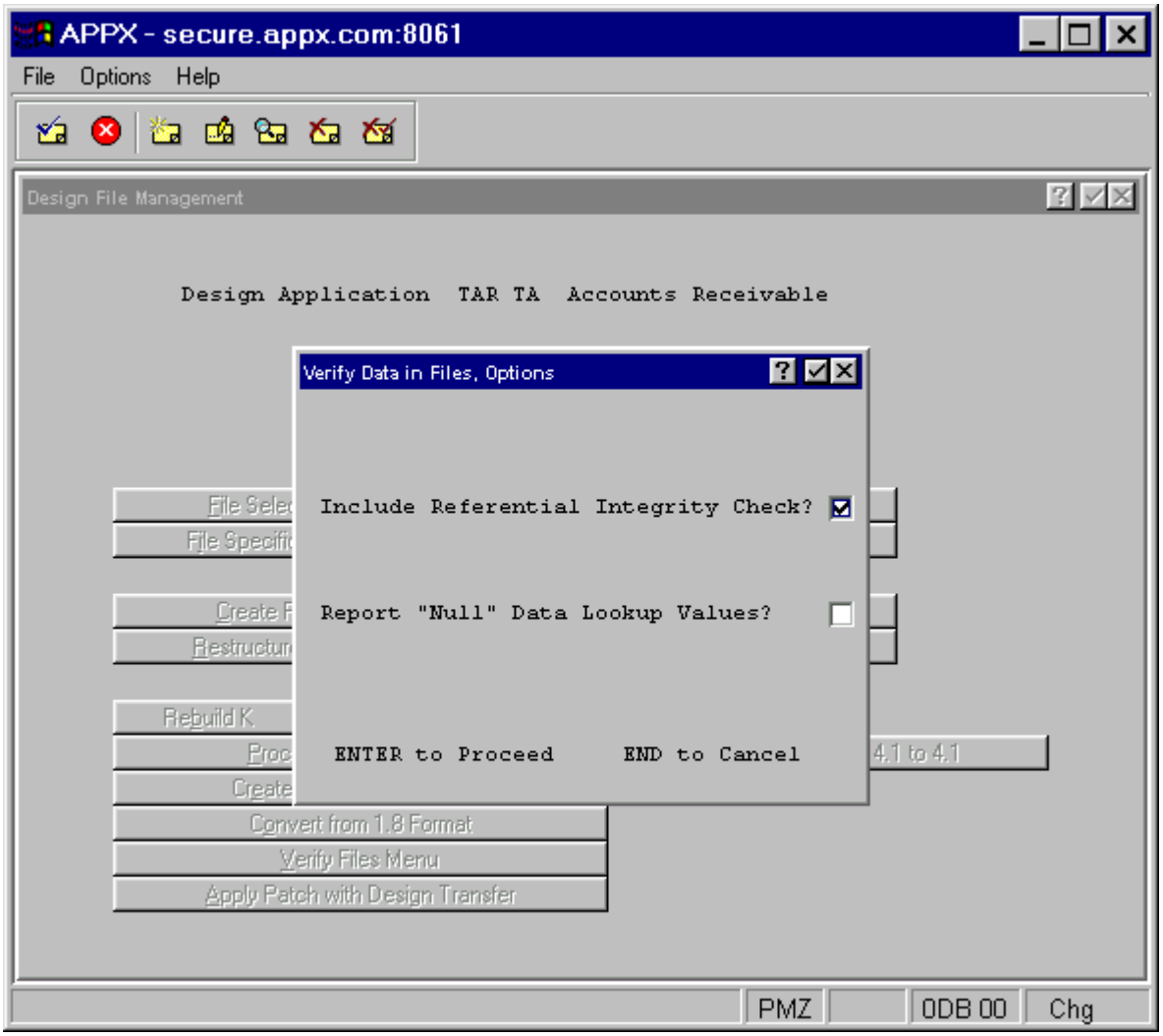


Figure 2-39. Verify Data In Files Options Overlay

SECURITY

APPX Security

APPX security assists in protecting sensitive information that is stored on your system. You can deny access to any field in an APPX-based system. There are multiple ways of protecting fields; you can protect them at the domain (field template) level, at the file level, or by individual field or work field. Additionally, you can prevent a user from executing any process. You can prevent a user from adding, deleting, or changing records in a file. Also, you can define a Record Access Security field that prevents a user from even knowing that a particular data record exists.

You accomplish this by assigning a security code to a domain, file, field, or work field, or to a process or set of processes, or to a set of records in a particular file. A *security code* is a four-digit number from 0000 to 9999. Rights to these codes are established in *Security Profiles*, and each user on your system can be assigned to a Security Profile. Security profiles can cover all databases or be assigned to a particular database.

There are many ways to set up Security Profiles as a means of managing field, record, and process protection on your system. For example, Security Profiles can be set up for “Management” and “Clerical”, or for “Accounts Payable” and “Accounts Receivable”, or even for “Security Management” if you wish. If needed, you can set up a separate Security Profile for a given user, although most users can be placed into general categories. You should spend some time planning your security system before setting up and assigning Security Profiles. You will want security to be easily maintained.

In order to secure your system effectively, several steps are necessary:

- Set up Security Profiles that define the security codes that will protect various areas of your system.
- Assign each user to an appropriate Security Profile.
- Enter security codes for the domains, files, fields and work fields you want to protect according to the hierarchy established by your Security Profiles.
- Enter security codes to protect processes or sets of processes, both for execution and for capability to add, delete, or change records in files.
- For files that contain individual records you want to secure, establish Record Access Security fields and design input processes that allow you to maintain a security code for each record you want to protect.

The remainder of this chapter describes how to set up Security Profiles and how to define what each user will be allowed to see or maintain within your system. Another chapter in this manual, Application Security Management, describes how to enter security codes to protect domains, files, fields, work fields, and processes. In addition, the *Designer Reference Manual* contains information regarding the establishment of Record Access Security. Thus, the implementation of security is a joint responsibility of the system administrator and the application designer.

Security Menu

The SECURITY option on the System Administration menu displays the Security menu shown in *Figure 2-40* below.

Security menu

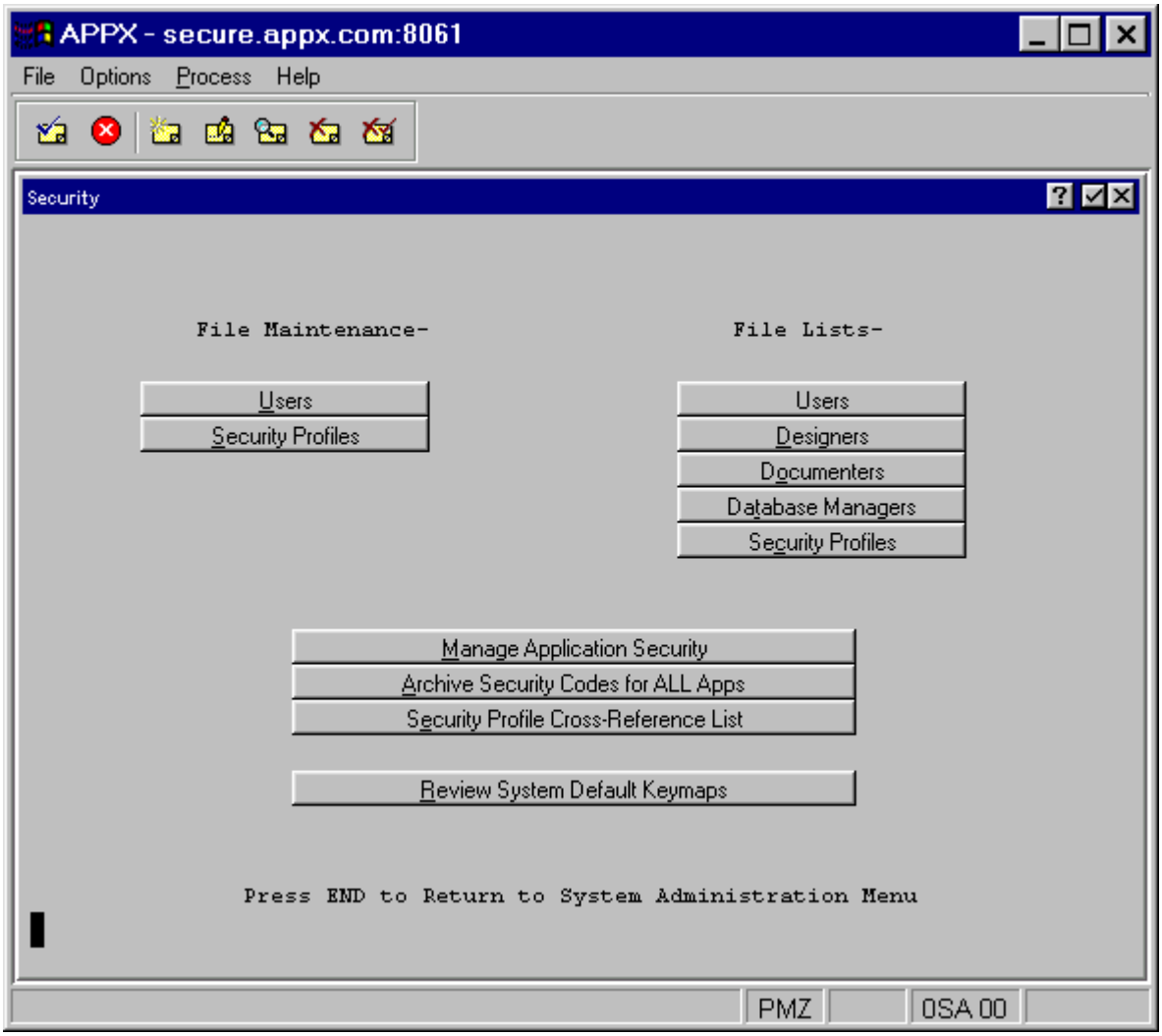


Figure 2-40

The options available on the Security menu are discussed below.

File Maintenance

Each of the file maintenance options is described later in this manual. See Users File Maintenance and Security Profiles File Maintenance for details.

File Lists

Options are available for printing the complete contents of the Users and Security Profiles files. You may also print limited listings of the Users file by selecting the options for DESIGNERS, DOCUMENTERS, or DATABASE MANAGERS.

Manage Application Security

This option accesses the Manage Application Security menu of APPX which is described in *Chapter 5* of this manual. Application Security Management can also be accessed directly from the APPX main menu.

Archive Security Codes for ALL Apps

This option is described in the Archive Security Codes for ALL Apps section of this manual. A similar option is available for the currently selected application from the Utilities menu within Application Security Management. See Security Management Utilities for details.

Security Profile Cross-Reference List

This option is described in the Security Profile Cross-Reference List section of this manual.

Review System Default Keymaps

This option is described in the Review System Default Keymaps section of this manual.

Users File Maintenance

You use Users File Maintenance from the Security menu to identify all of the people at your site who are allowed to enter and run APPX. All types of APPX users, including system administrators, database administrators, application designers, and end users, must be entered in this file. The Users File Maintenance screen is shown in *Figure 2-41* below.

Users File Maintenance screen

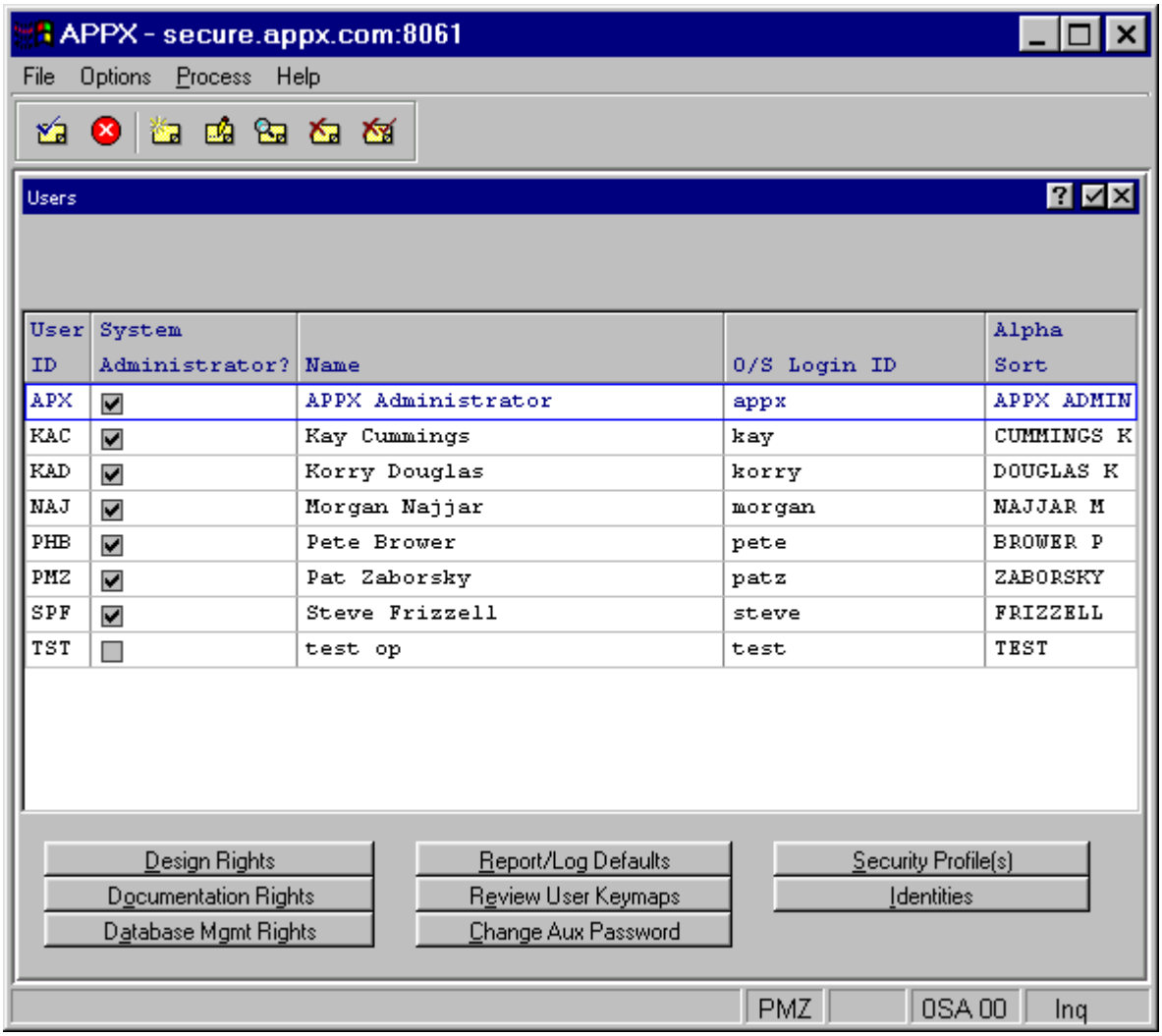


Figure 2-41

The Users File Maintenance screen contains the following fields:

User ID

Specifies the unique identifier of each user. You can, for example, use end users' initials.

System Administrator?

Designates this user as an APPX system administrator if set to "yes" ☑. The default value is "no" ☐ (the user is not an administrator).

Name

Allows you to specify the name that is associated with this user ID.

O/S Login ID

Specifies the internal system name that is assigned to the user. Depending on your computer system, this value may need to be all uppercase letters. Refer to your installation guide for specifics.

Alpha Sort

Specifies a mechanism for sorting the Users file.

This screen also contains the following options (see the specific overlay section for details):

- ***Design Rights*** identifies the applications for which the user has design authority.
- ***Documentation Rights*** identifies the applications whose documentation the user may modify.
- ***Database Mgmt Rights*** identifies the databases/applications which the user may administer.
- ***Report/Log Defaults*** allows for specific defaults for this user.
- ***Review User Keymaps*** lists all keymaps defined within APPX for this user.
- ***Change Aux Password*** defines a password for the user for processes outside of APPX.
- ***Security Profiles*** accesses Security Profiles File Maintenance constrained to the user's profile.

Identities

Are used to store login information (User ID, realm, and password) for various external interfaces such as an RDBMS. An identity record can exist for each FMS Group/User combination. See the Identities section for an explanation of how identities are used and the hierarchy in which they are employed. See the Identities Overlay for a selected user for field information. A similar Identities Overlay for a selected FMS group is available in File System Groups File Maintenance. Both overlays provide access to the same login information.

When you select a user and press ENTER, the Users File Maintenance overlay appears. This overlay is described in the next section.

Users File Maintenance Overlay

After pressing ENTER from the Users File Maintenance screen, the Users File Maintenance overlay appears as shown in *Figure 2-42* below.

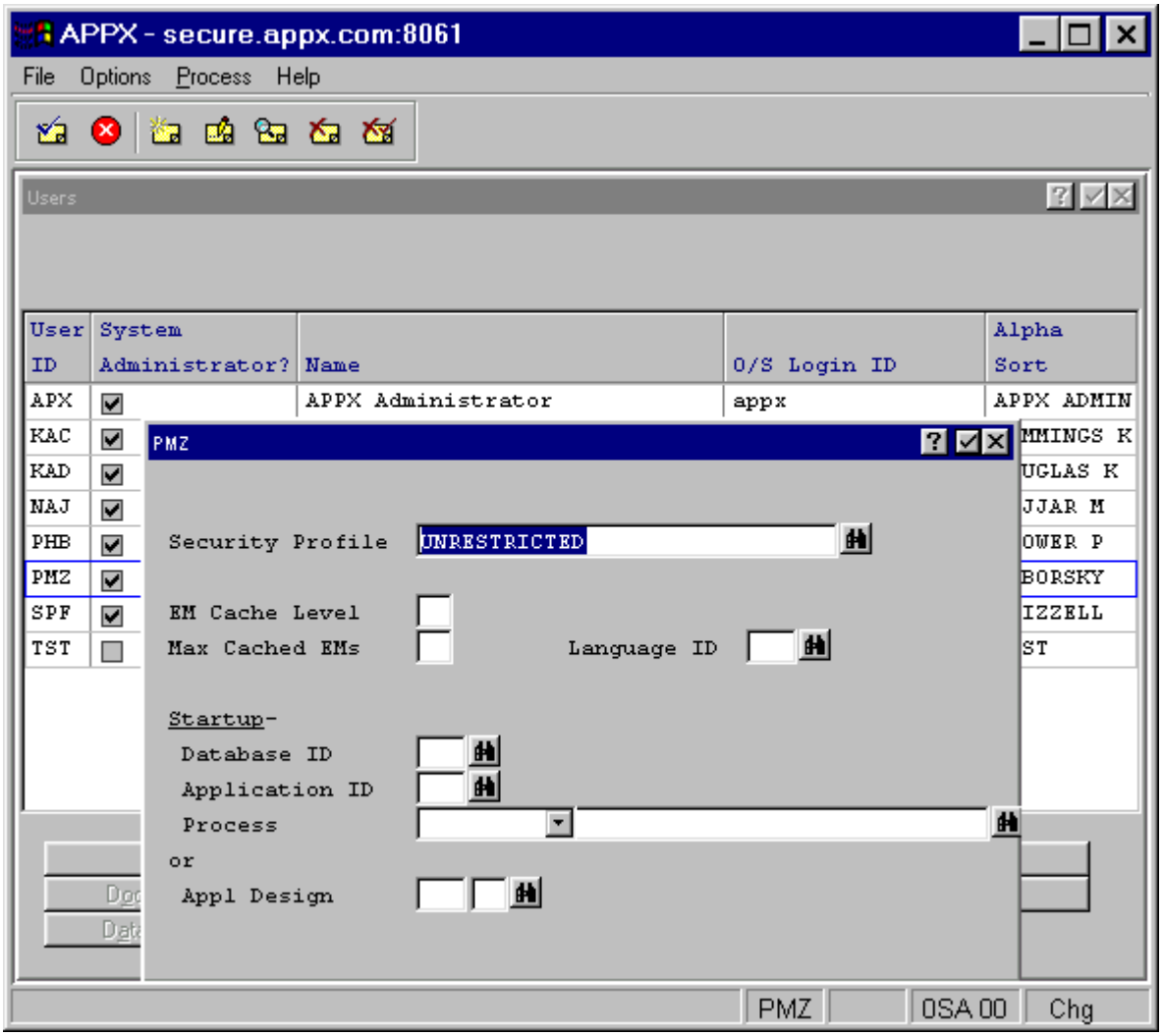


Figure 2-42. Users File Maintenance Overlay

This overlay allows you to enter additional information about the user as follows:

Security Profile

Associates this user with one of the valid security profiles that you define using Security Profiles File Maintenance. If blank, the user has rights to all security codes.

EM Cache Level

Is the user's "automatic discard level" for executable modules (EMs). Any process with a priority that is less than or equal to this number is never cached. The cache level ensures that low-priority EMs do not waste space in memory. If this field is blank, APPX uses standard values.

Max Cached EMs

Determines the maximum number of executable modules for inactive processes that can be retained in memory at the same time. When this limit is reached, APPX does not prohibit a new EM from entering the cache as long as its priority is greater than that of at least one EM already in the cache. The EM with the lowest priority is discarded. If this field is blank, APPX uses standard values.

Language ID

Represents the language in which all applications should be presented to this user. Any Language ID entered here overrides any other Language ID designated at the system, application or database level. This field should be left blank if the Language Translation feature is not being used. Select the *Scan* icon for a list of available languages.

Startup

Allows you to specify the initial process that automatically invokes when the user enters APPX:

Database ID

Identifies the database in which the startup process will be invoked.

Application ID

Identifies the application that contains the startup process.

Process

Identifies the type and name of the process that will be invoked at startup.

- or

Appl Design

Identifies the application and version that will be automatically defaulted when the designer enters Application Design.

Users Design Rights Overlay

The Users Design Rights overlay accessed from the Users File Maintenance screen is shown in *Figure 2-43* below. It identifies all the applications in which the selected user is entitled to perform application design.

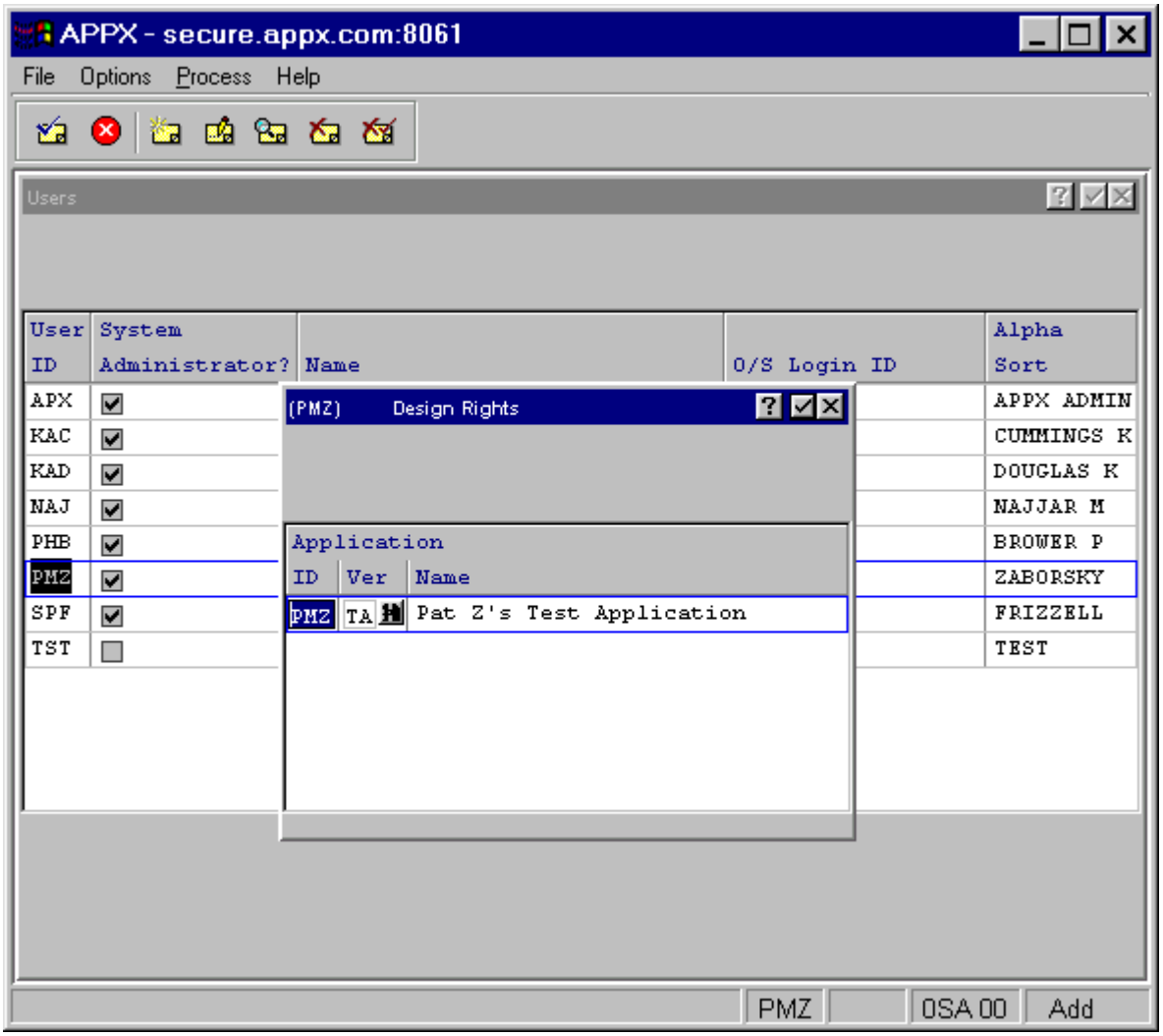


Figure 2-43. Users Design Rights Overlay

The Users Design Rights overlay contains the following fields:

Application ID/Ver

Identifies each valid application and version that this user has design authority for. If you leave the version blank, the designer can access all versions of the designated application; if you leave both of these fields blank, the designer has authorization to access all applications and versions at your installation.

Users Documentation Rights Overlay

The Users Documentation Rights overlay accessed from the Users File Maintenance screen is shown in *Figure 2-44* below. It identifies all the applications for which the user may enter and/or edit documentation.

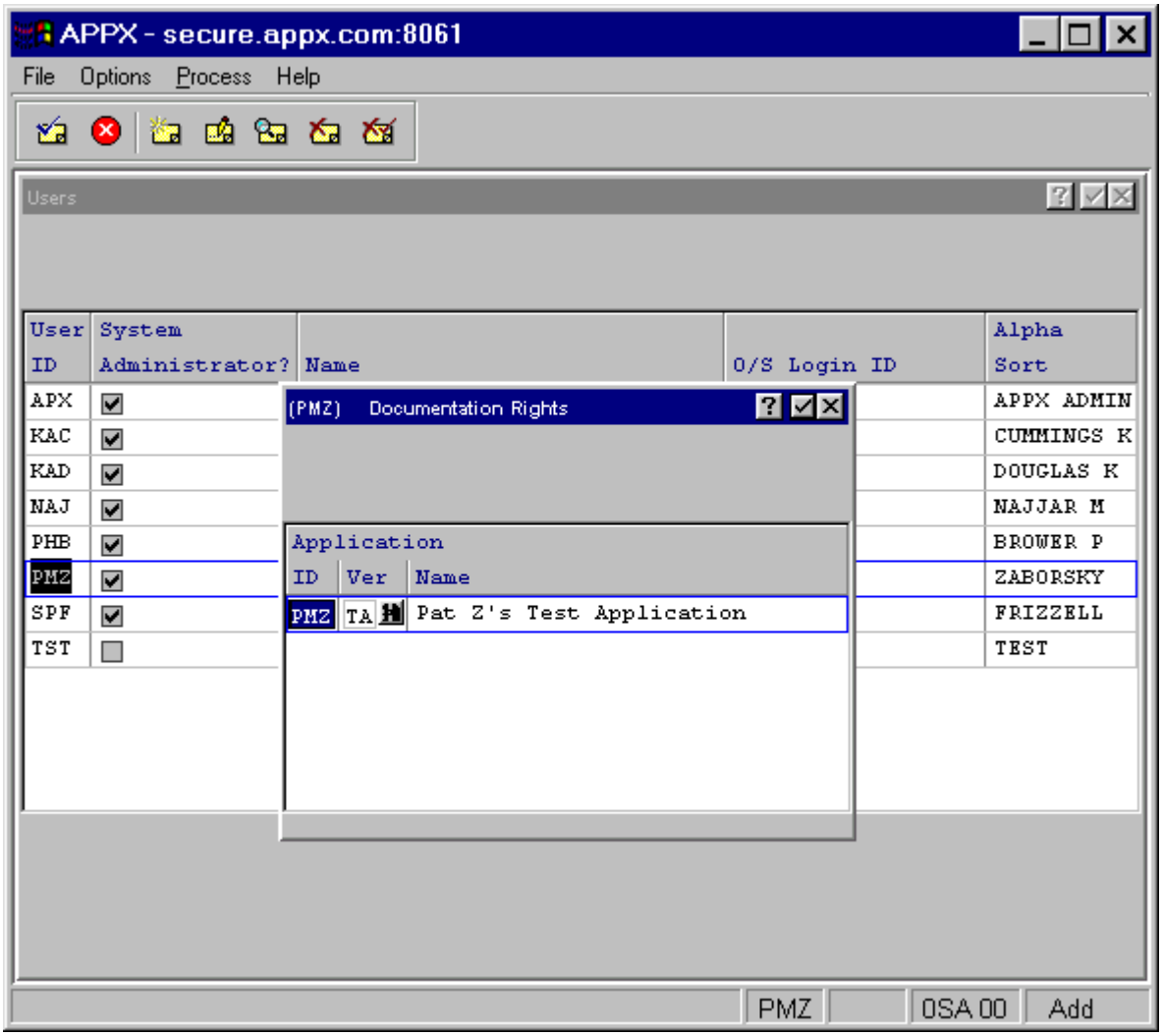


Figure 2-44. Users Documentation Rights Overlay

The Users Documentation Rights overlay contains the following fields:

Application ID/Ver

Identifies each valid application and version that this user has documentation authority for. If you leave the version blank, the designer can access all versions of the designated application; if you leave both of these fields blank, the designer has authorization to access all applications and versions at your installation.

Users Database Mgt Rights Overlay

The Users Database Management Rights overlay accessed from the Users File Maintenance screen is shown in *Figure 2-45* below. It identifies databases and/or applications for which the selected user may act as database administrator at your site.

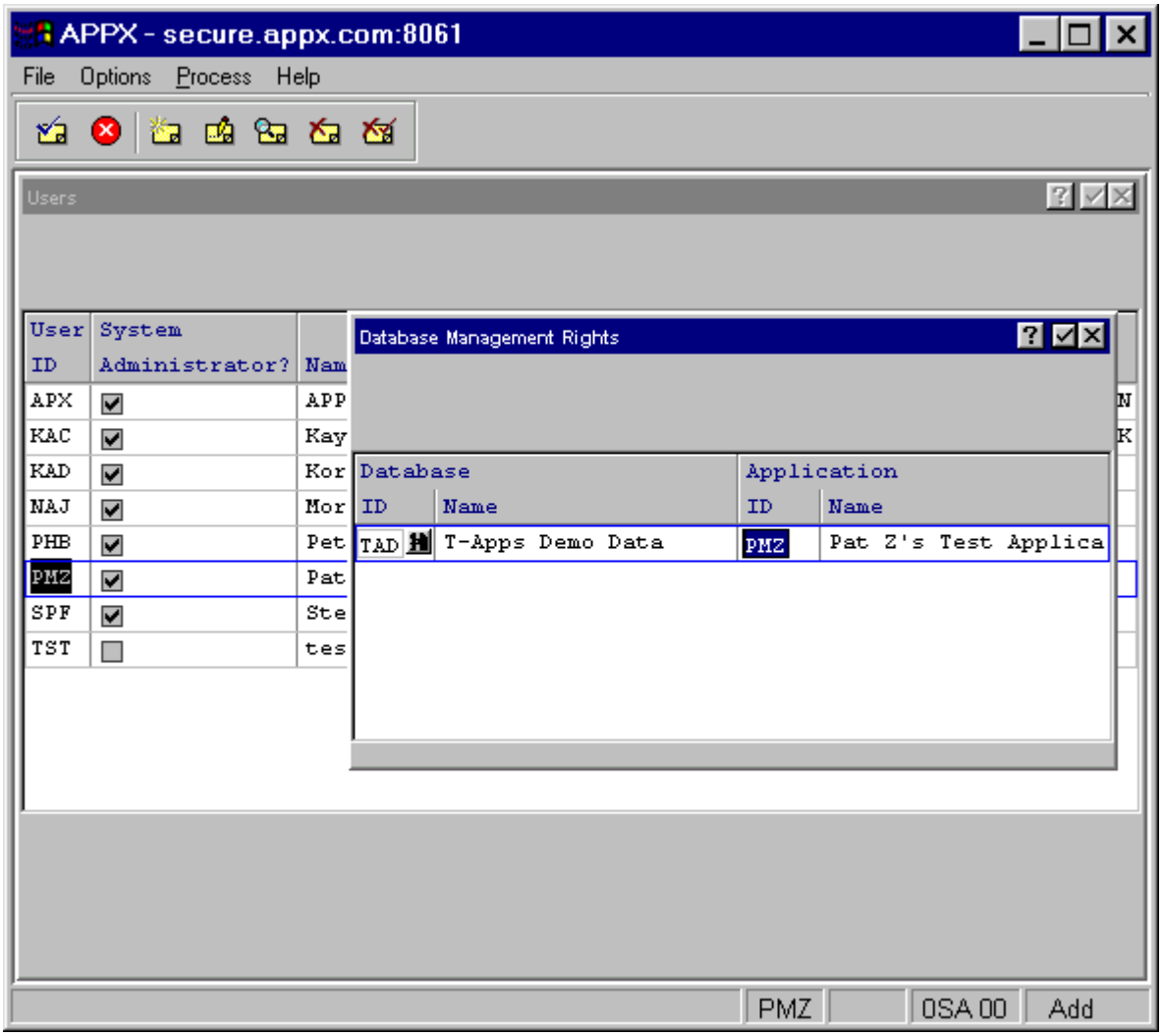


Figure 2-45. Users Database Management Rights Overlay

The Users Database Management Rights overlay contains the following fields.

Database ID

Specifies a valid database that this person has administration privileges for. If this field is blank and *Application ID* is specified, the user is authorized to manage this application for any database.

Application ID

Specifies a valid application that this person has database administration privileges for. If this field is blank, the user is authorized to manage all applications for the specified database.

> **Note:** If *Database ID* and *Application ID* are both blank, the user is authorized to act as database manager for all databases and all applications.

Users Report/Log Defaults Overlay

If you select the REPORT/LOG DEFAULTS option from the Users File Maintenance screen, the Users Report/Log Defaults overlay appears as shown in *Figure 2-46* below. All entries on this screen are optional. If no value is specified, the APPX run time default hierarchy applies. For field details, refer to the System Parameters Report/Log Defaults overlay which is identical to this overlay.

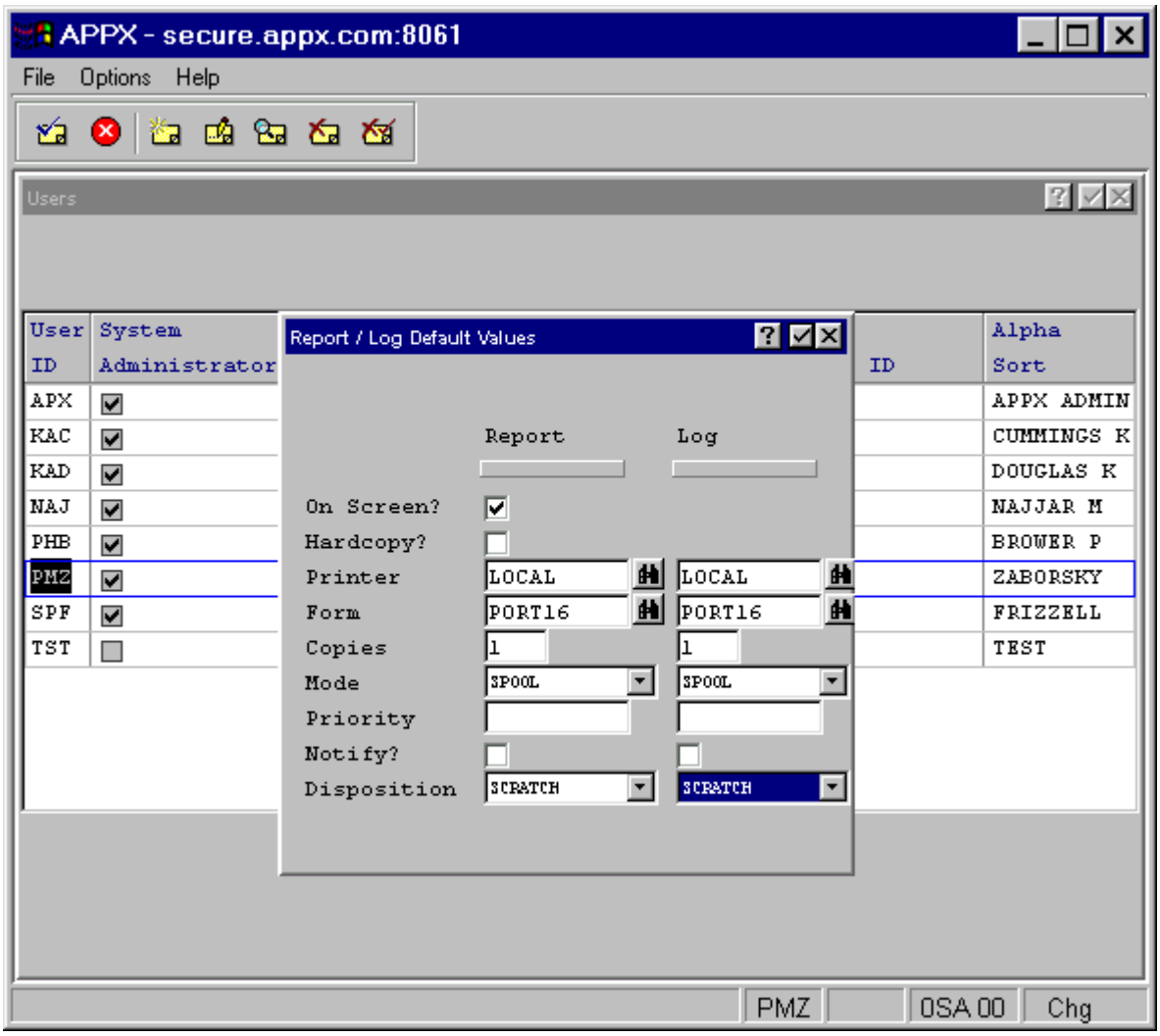


Figure 2-46. Users Report/Log Defaults Overlay

Review User Keymaps

The Review User Keymaps overlay accessed from the Users File Maintenance screen is shown in *Figure 2-47* below. It lists all keymaps defined within APPX for the selected user. The COPY A KEYMAP option allows you to copy the selected keymap to another user/keymap combination. When copying, you may use User --- to represent the system-wide default keymap for all users.

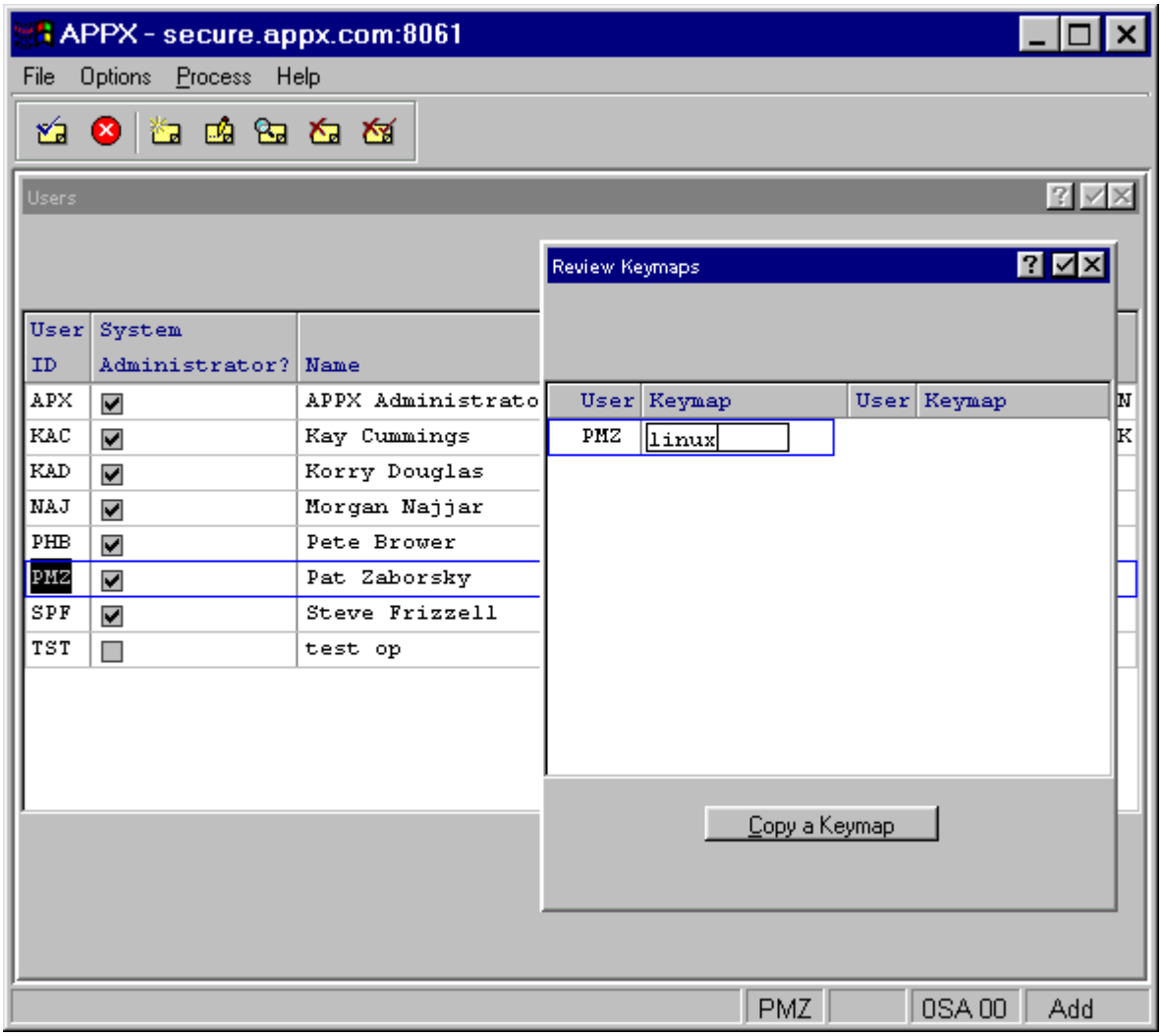


Figure 2-47. Review User Keymaps Overlay

To see all the default keymaps available within APPX regardless of user, select the REVIEW SYSTEM DEFAULT KEYMAPS option from the Security menu.

Change Auxiliary Password

The Change Auxiliary Password overlay accessed from the Users File Maintenance screen is shown in *Figure 2-48* below. It is used to define a new password for the selected user when running processes outside of APPX, such as connecting to an external RDBMS. After entering the new password and confirming it, select CHANGE PASSWORD to finalize the change.

See the Identities section for the default hierarchy that determines when this password will be used.

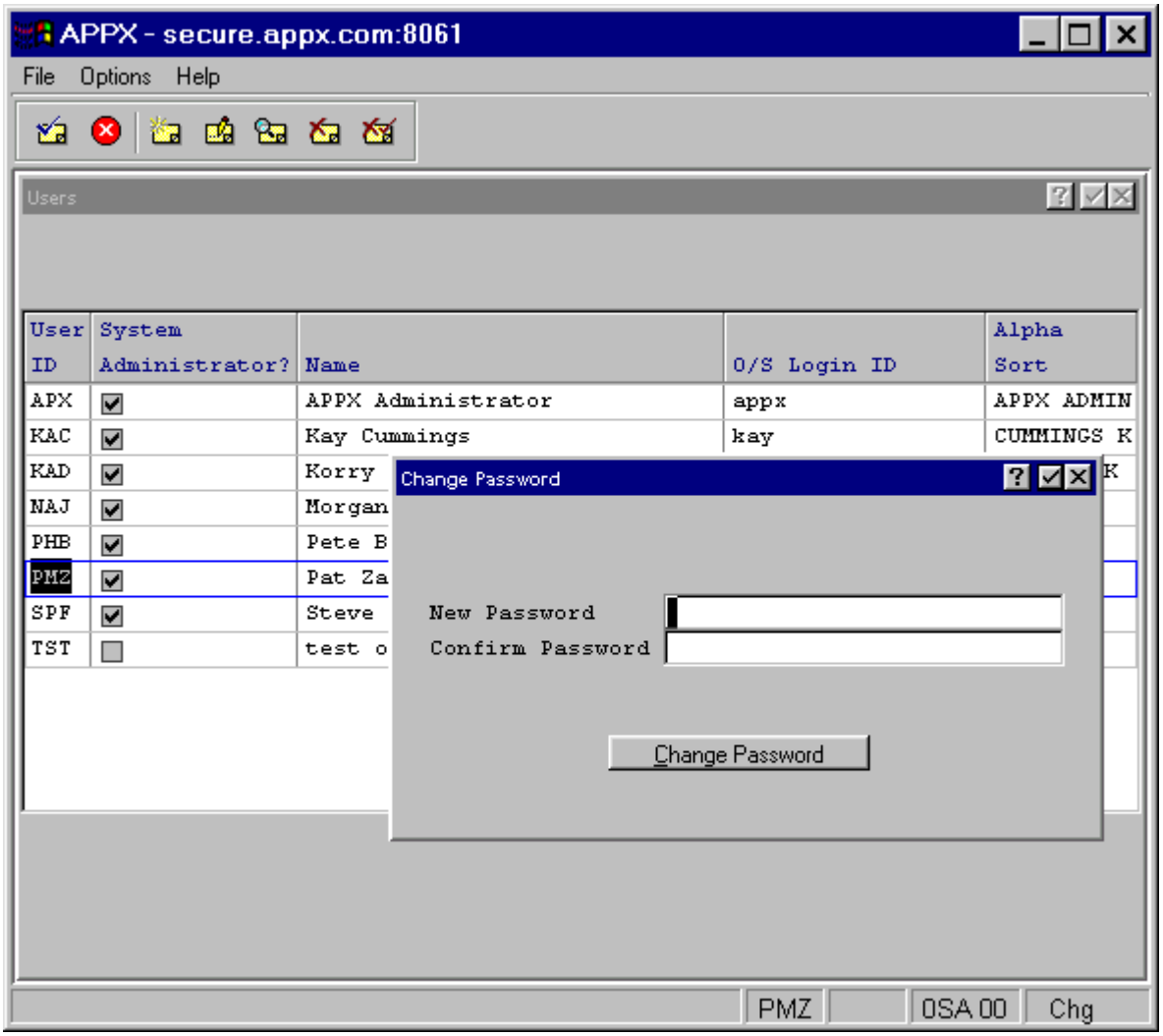


Figure 2-48. Change Auxiliary Password Overlay

Users Security Profiles Overlay

The Users Security Profiles overlay accessed from the Users File Maintenance screen is shown in *Figure 2-49* below. It runs Security Profiles File Maintenance but constrains it to the user's security profile (which was specified in the Users File Maintenance overlay). This allows you to view and/or edit security rights for that profile and its associated databases. Note, however, to set up new security profiles or to avoid constraints, you should run Security Profiles File Maintenance from the Security menu.

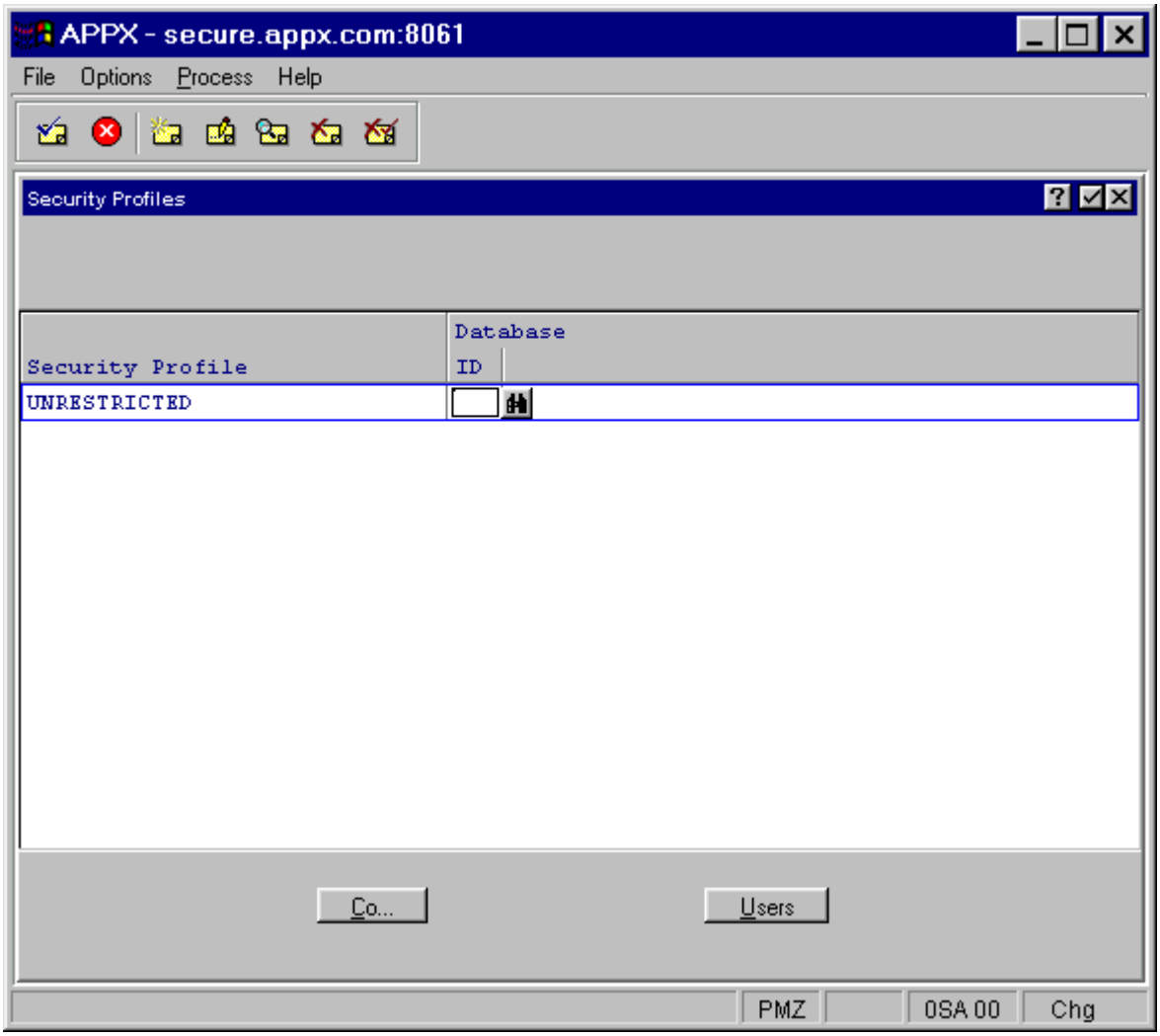


Figure 2-49. Users Security Profiles Overlay

The Users Security Profiles overlay contains the following items:

- Security Profile identifies the security profile currently specified for this user.
- Database ID identifies a valid database associated with this security profile. When a user that has a security profile selects a database, APPX first tries to find a record for the specified profile/database combination. If the combination cannot be found, APPX uses the profile with a blank database ID.

See Security Profiles File Maintenance for a complete description of the options and overlays associated with this screen.

Users Identities Overlay

The Users Identities overlay accessed from the Users File Maintenance screen is shown in *Figure 2-50* below. It is used to establish an "identity" for storing login information needed for various external resources. (See the Identities section for an explanation of how identities are used.) The login

information is stored by FMS Group/User combination and can be accessed from this overlay or from the Identities overlay within File System Groups File Maintenance. You can add, delete, or change this information in either place.

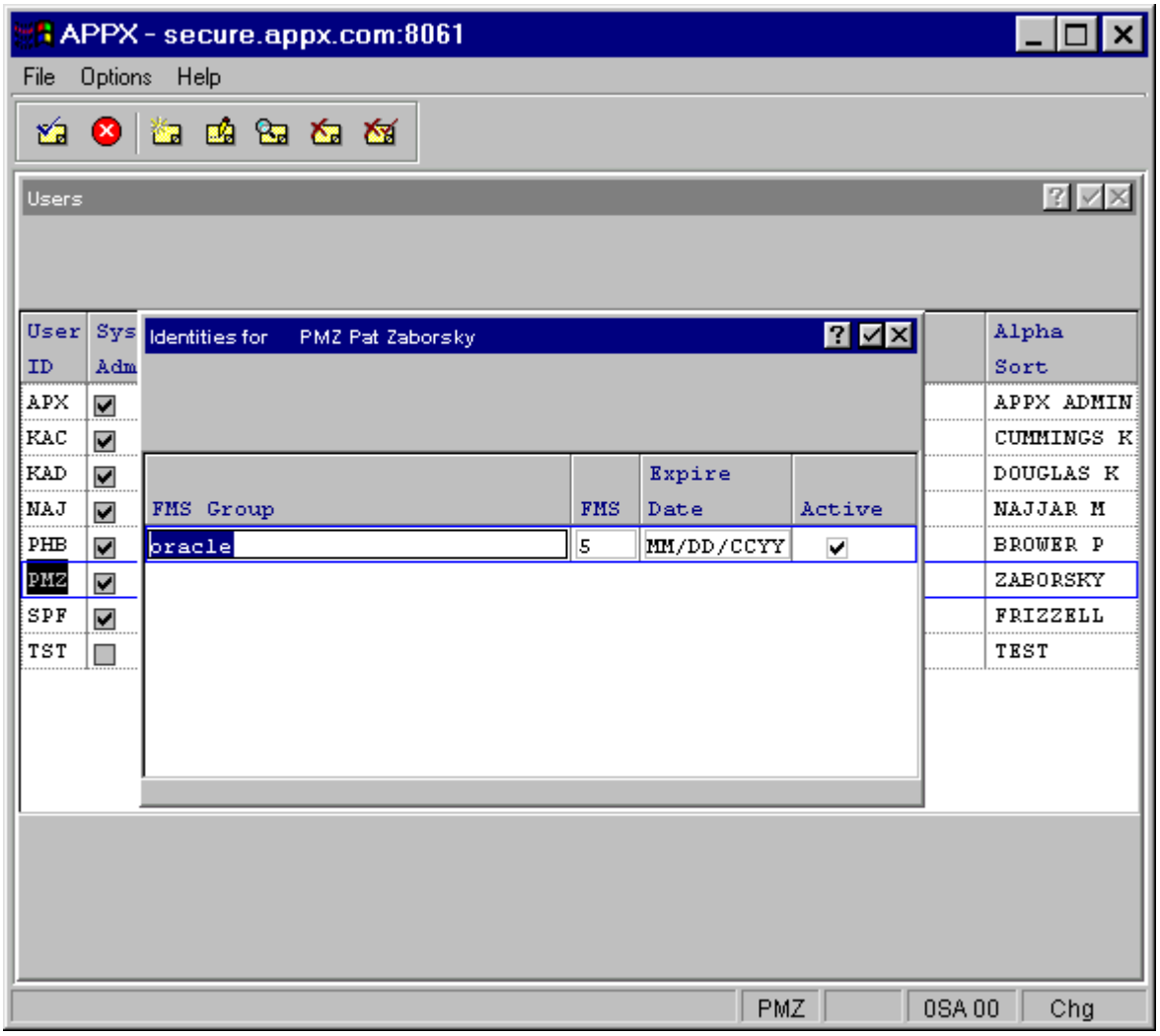


Figure 2-50. Users Identities Overlay

The Users Identities overlay contains the following fields:

FMS Group

Specifies the name of a file management system (FMS) group.

FMS

Specifies the type of ISAM (Indexed-Sequential Access Method) used in the file management system. See File System Groups File Maintenance for a list of valid FMS values.

Expire Date

Specifies the date this identity is no longer valid.

Active?

Can be toggled between “yes” ☑ and “no” ☐ to indicate whether this identity is active or not. This allows you to retain inactive identities that may be needed again in the future.

When you press ENTER from this screen, an overlay appears which allows entry of User ID, realm, and password. See the Identities Detail overlay within File System Groups File Maintenance for more information.

Security Profiles File Maintenance

As described in the APPX Security section, APPX uses a system of security codes and security profiles to assist in protecting sensitive information stored on your system.. A *security code* is a four-digit number from 0000 to 9999, the rights to which are established in security profiles. A *security profile* contains the rights associated with a specific set of 10,000 security codes numbered from 0000 to 9999. These rights can be granted to one or more users at your installation by assigning the security profile to the user in the Users File Maintenance overlay. There can be any number of security profiles. In addition, a security profile name can be associated with specific database(s) and/or with a blank database which will serve to cover all other databases for that profile. Each security profile name/database combination can contain a different set of rights.

Security Profiles File Maintenance is used to establish and maintain security profiles. The first screen specifies the name of the security profile and its associated database, if any. Following screens are overlays in which you specify what rights or privileges are included in the security profile. *Figure 2-51* is an example of the first screen in Security Profiles File Maintenance:

Security Profiles File Maintenance screen

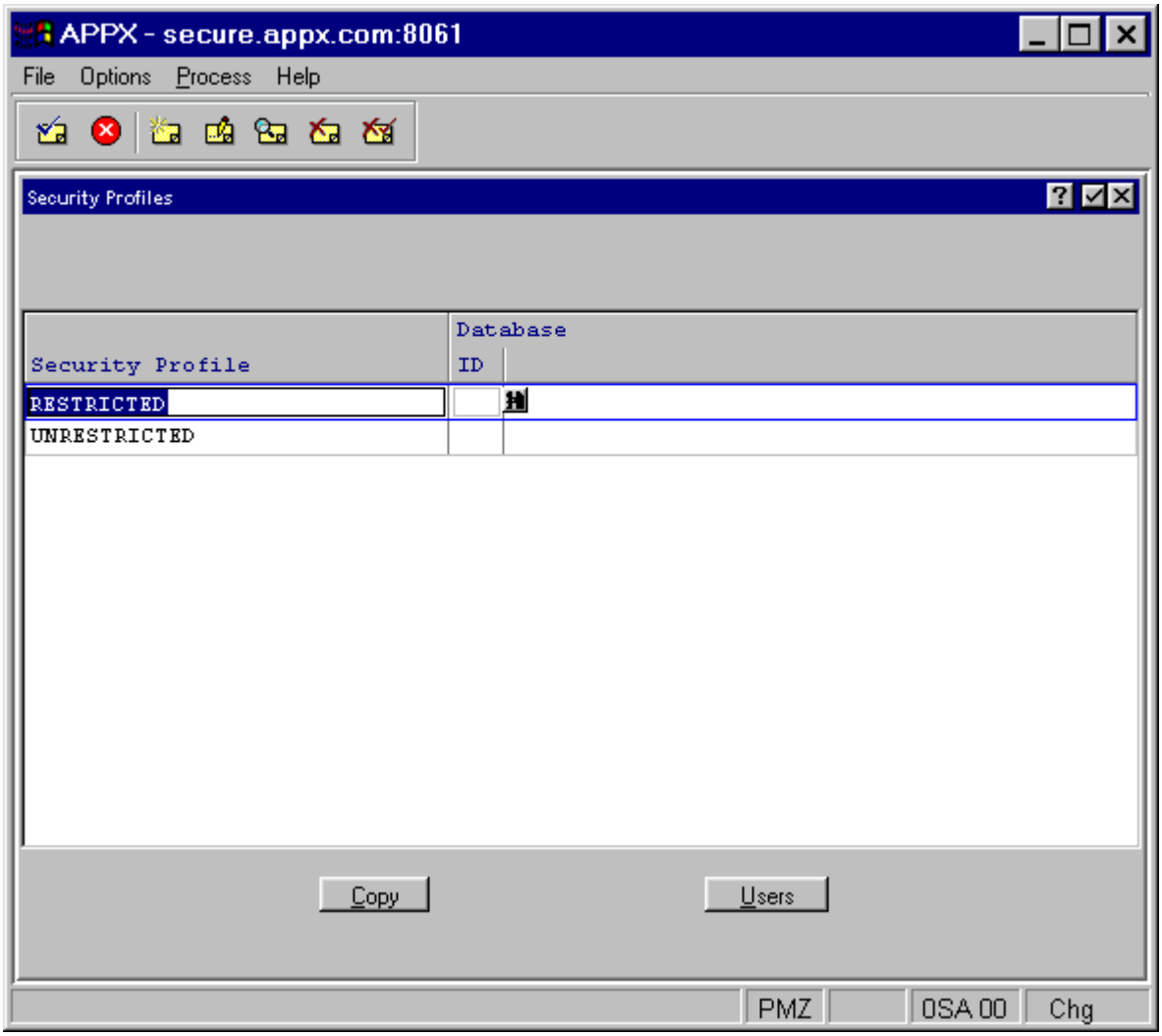


Figure 2-51

The Security Profiles File Maintenance screen contains the following fields:

Security Profile

Identifies the name of the security profile. Any profile can be set up with a blank database ID, and also in association with one or more valid databases. The profile associated with a blank database may be used for any database. If the same profile exists with a valid database, that profile overrides any values set up in the profile/blank database record, for that database only. If a user has been given a security profile, and attempts to access a database, APPX will look first for a profile/database combination that matches the user's profile and the selected database. If found, the user will be allowed access to any data and process objects which match the selected profile. If no such profile/database record exists, APPX will then look for a profile/blank database record matching the user's profile. If found, the user will be allowed access to any data and process objects which match that profile. If no such profile/database record exists, APPX will deny the user access to any data or process object containing a security code.

Database ID

Identifies a valid database that you want this security profile to be associated with. This profile will take precedence over the same profile with a blank database ID, for this database.

The Security Profiles File Maintenance screen contains two options:

Copy

Allows you to copy the security rights of another security profile to the current security profile. See the Security Profiles Copy Rights Table overlay for details.

Users

Displays a list of users currently assigned to this profile.

When you select a profile and press ENTER, the Security Profiles File Maintenance overlay appears. This overlay is described in the next section.

Security Profiles F/M Overlay

When you press ENTER from the Security Profiles File Maintenance screen, the overlay shown in *Figure 2-52* below allows you to specify rights to any of the 10,000 security codes associated with the currently selected security profile/database combination.

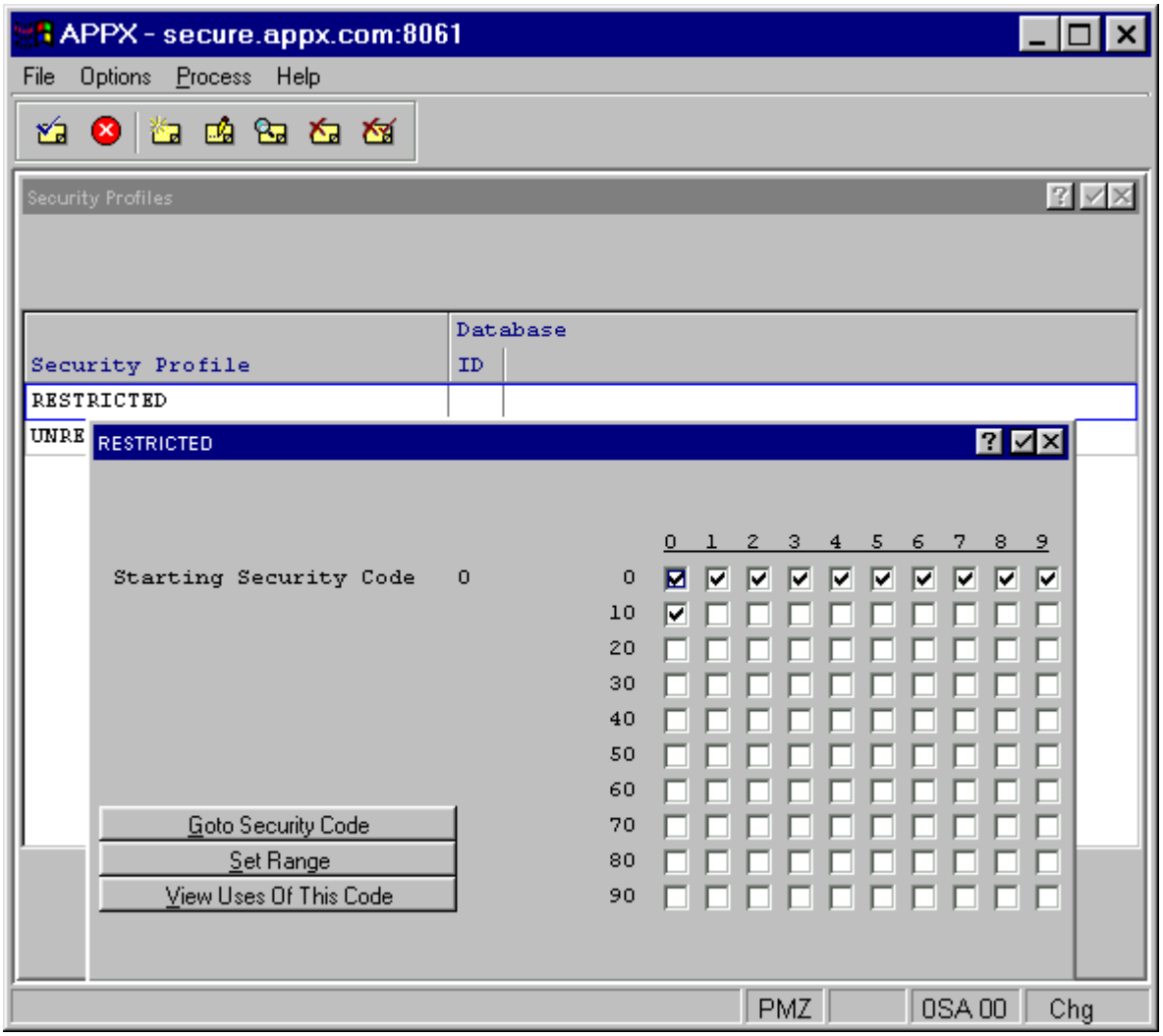


Figure 2-52. Security Profiles File Maintenance Overlay

This overlay contains the following fields:

Starting Security Code

Is a multiple of 100 from 0 to 9,900 that is supplied by the system. Use the GOTO SECURITY CODE option below to modify the starting point.

- ***Security Code Table*** At this point, the overlay display contains a matrix table of 100 security codes, each with a yes/no checkbox. The column along the left side of the screen lists a multiple of 10 (for example, 0, 10, 20, etc.) for 10 groups, starting at the number you previously designated. The right side of the screen contains rows of yes/no checkboxes, with column headings to indicate each unit position (0, 1, 2, etc.). In our example, the first row contains codes 0, 1, 2, etc., through 9. The second row starts at 10 and goes through 19. This pattern continues through each row, and the last code on the screen is 99. For each code, you can select "yes" ☑ or "no" ☐, indicating whether or not this security profile/database combination has rights to this code. Click the checkbox to toggle between “yes” and “no”.

This overlay contains the following options:

Goto Security Code

Allows you to specify which security code you want to start with in the security code table. See the Goto Security Code overlay section for details..

Set Range

Allows you to set a new value for the security codes within a specified range. See the Set Range overlay for details.

View Uses Of This Code

Generates a report of processes, files, and fields in which the security code your cursor is currently positioned on is used. All applications are included. Note that security codes must have been archived prior to selecting this option. For more information on this process, see the Archive Security Codes for ALL Apps section later in this chapter.

Goto Security Code Overlay

The Goto Security Code overlay accessed from the Security Profiles File Maintenance overlay is shown in *Figure 2-53* below. It provides for modifying the starting value of the Security Code Table to be displayed on the screen.

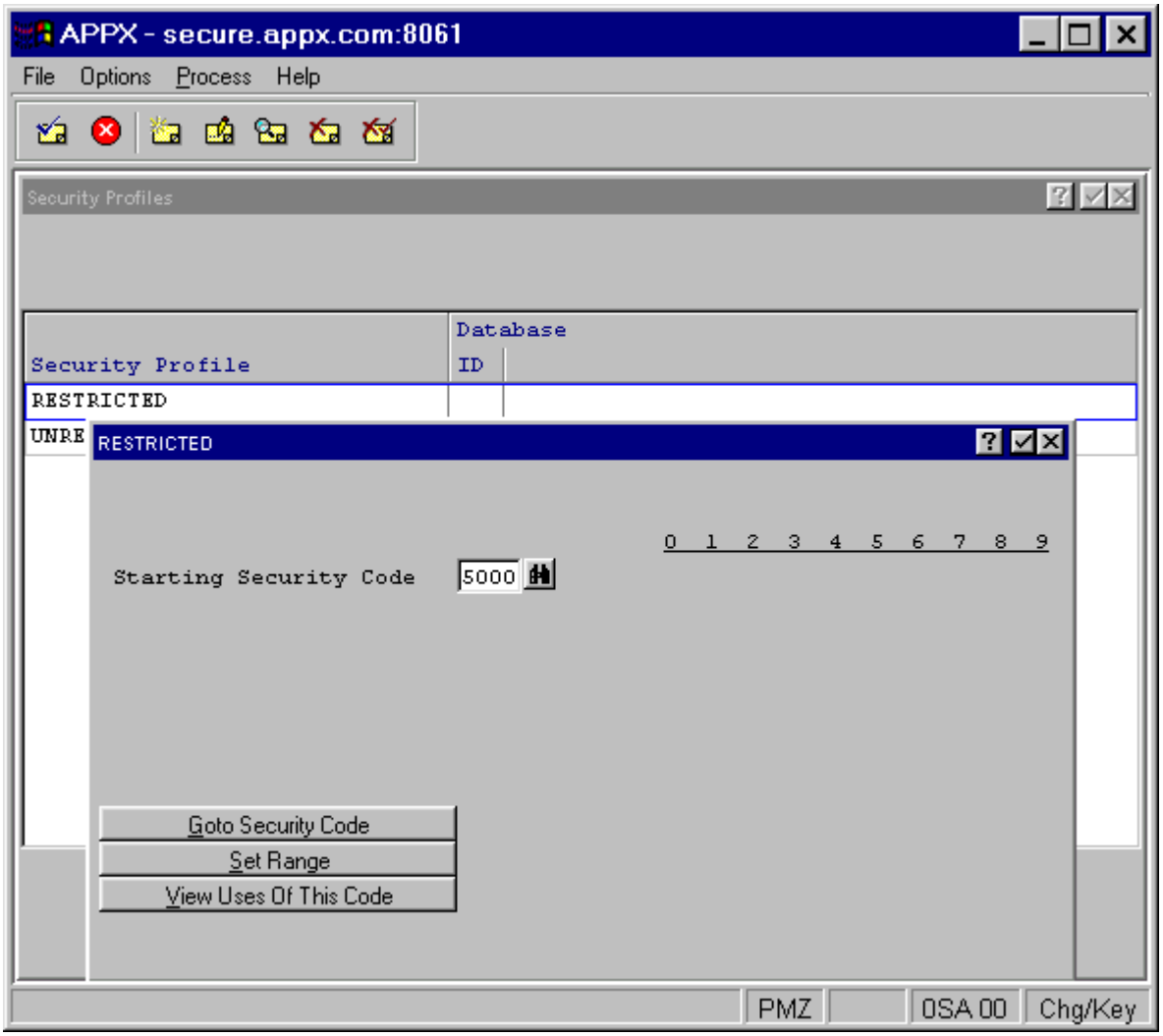


Figure 2-53. Goto Security Code Overlay

The Goto Security Code overlay contains the following fields:

- *Starting Security Code* Enter the security code where you want to begin defining yes/no values. Pressing ENTER then redisplays the Security Code Table, beginning with the hundreds number that includes the number you entered. For example, if you enter 250, the redisplayed table will begin with 200.

Set Range Overlay

The Set Range overlay accessed from the Security Profiles File Maintenance overlay is shown in *Figure 2-54* below. It allows you to enter a new value for a specified range for the currently selected profile.

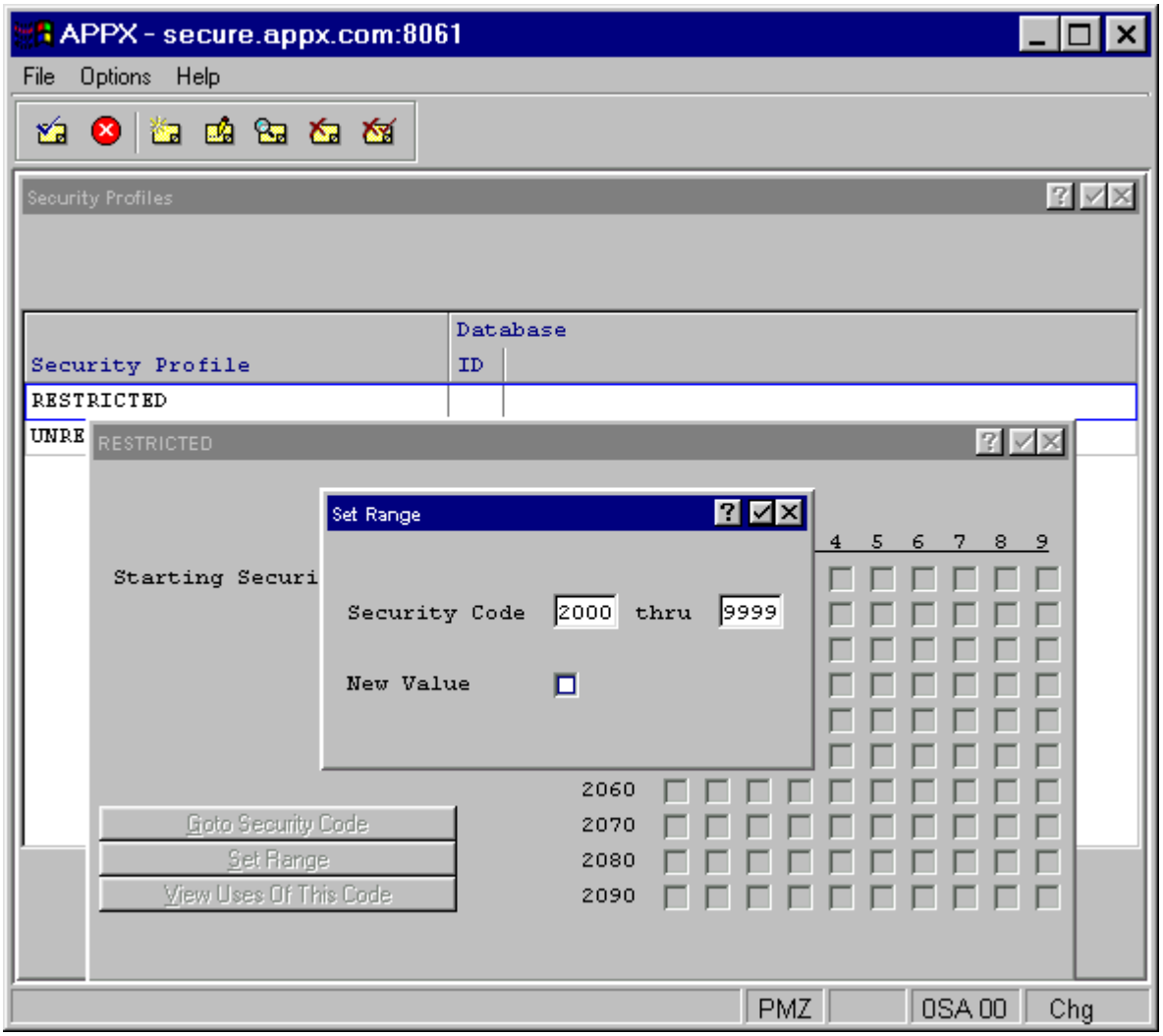


Figure 2-54. Set Range Overlay

The Set Range overlay contains the following fields:

- ***Security Code*** Enter the beginning security code of the range of codes you want to change...

---

thru

The ending security code of the range of codes you want to change.

- ***New Value*** Click the checkbox to toggle between “yes” ☑ and “no” ☐. This will be the value to which each code in the range will be set when you press ENTER. "Yes" means that the profile will have security rights to the codes within the range; “no” means that the profile will not have rights to these codes.

Copy Rights Table Overlay

When you select the COPY option from the Security Profiles File Maintenance screen, the Copy Rights Table overlay shown in *Figure 2-55* below appears. It provides for copying the security rights of another specified profile to the currently selected profile.

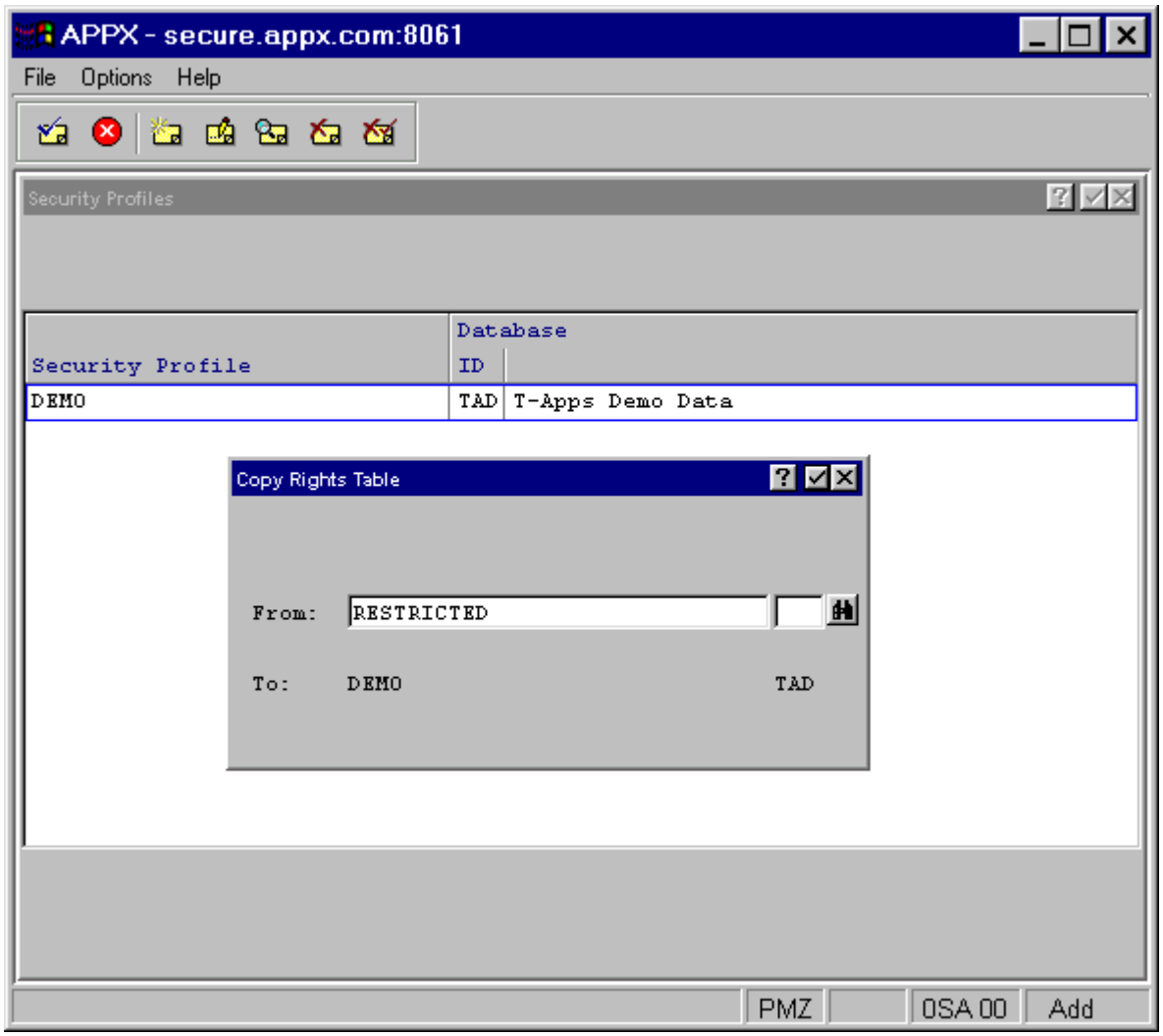


Figure 2-55. Copy Rights Table Overlay

This overlay contains the following field:

- ·· ***From:*** Enter the name and database ID of the security profile whose rights you wish to copy to the current profile (specified in the *To:* fields)

Archive Security Codes for All Apps

When you select the ARCHIVE SECURITY CODES FOR ALL APPS option from the Security menu, a screen appears as shown in *Figure 2-56* below:

Archive Security Codes for ALL Apps screen

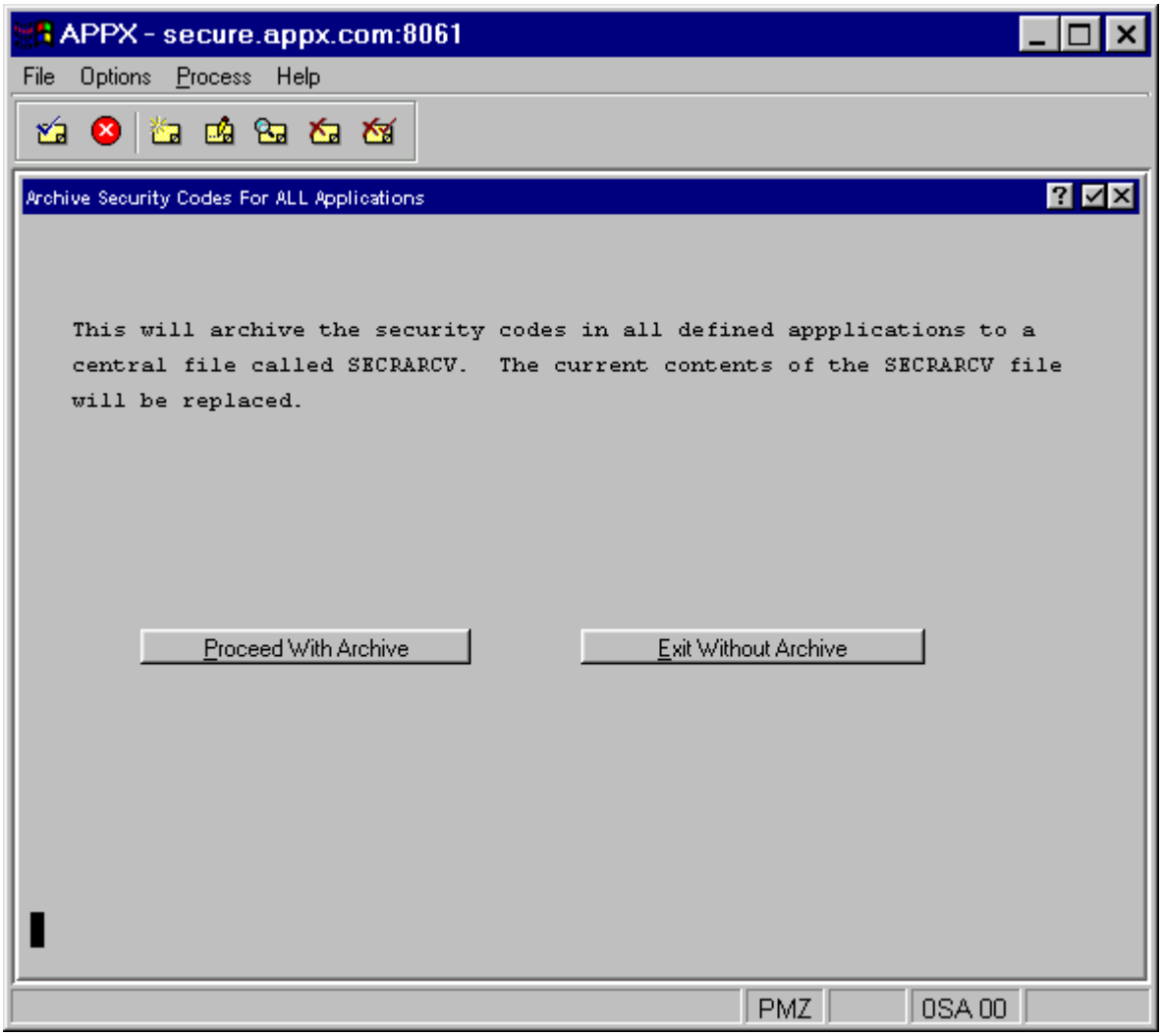


Figure 2-56

This option archives all security codes in all defined applications to a central archive file* found in System Administration. This archiving is required prior to using the VIEW USES OF THIS CODE option in Security Profiles File Maintenance or to running the Security Profile Cross-Reference List.

Since the current contents of the archive file will be lost when archiving begins, you must elect to proceed with the archive or to exit without archiving.

***Note:** The Archived Security Codes file (SECRARCV) is also used by Security Management Utilities found within Application Security Management, which include Security Lists, Clear Security Codes, Archive Security Codes, and Post Security Codes from Archive.

Security Profile X-Ref List

The SECURITY PROFILE CROSS-REFERENCE LIST option from the Security menu prints a report that cross-references the Security Profiles file (PROFILE) with the Archived Security Codes files (SECRARCV). When a security code is set to "yes" ☑ and records in the archive exist for that security code, a line is printed identifying the Data Dictionary element or Process that uses that security code. At the start of each Security Profile is a list of users with access to that Security Profile.

Prior to printing this report, security codes must have been archived by using the ARCHIVE SECURITY CODES FOR ALL APPS option either from the Security menu or from this screen as shown in *Figure 2-57* below. Once archived, you may proceed by clicking the SECURITY PROFILE XREF LIST option.

Security Profile Cross-Reference List screen

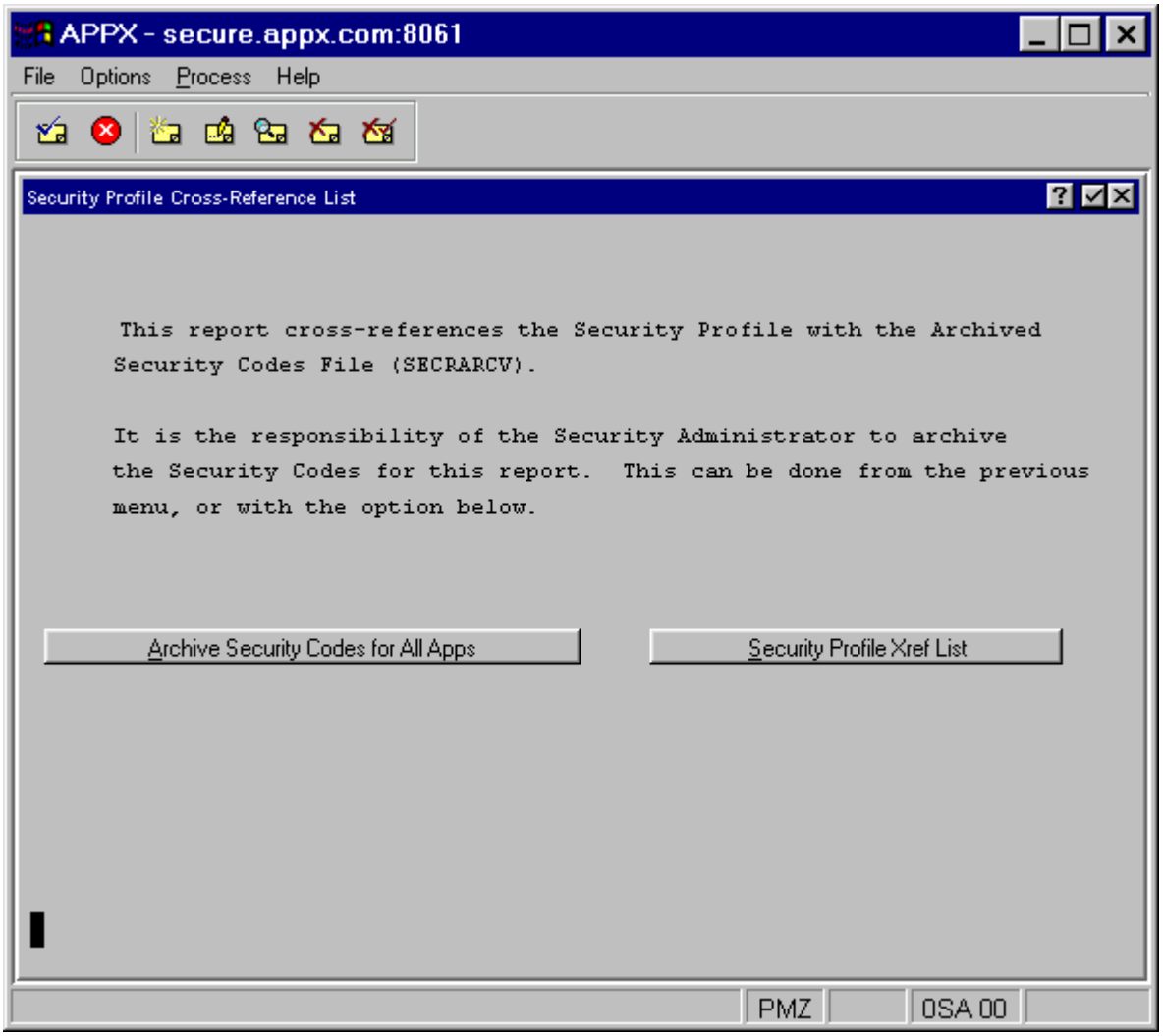


Figure 2-57

Review System Default Keymaps

The REVIEW SYSTEM DEFAULT KEYMAPS option from the Security menu lists all keymaps currently defined within APPX as shown in *Figure 2-58*. These keymaps can be system-wide (where User ID is ---) or assigned to an individual user. To only see the keymaps defined for a specific user, select the REVIEW USER KEYMAPS option available within Users File Maintenance.

The COPY A KEYMAP option allows you to copy the selected keymap to another user/keymap combination. When copying, you may use User --- to signify all users.

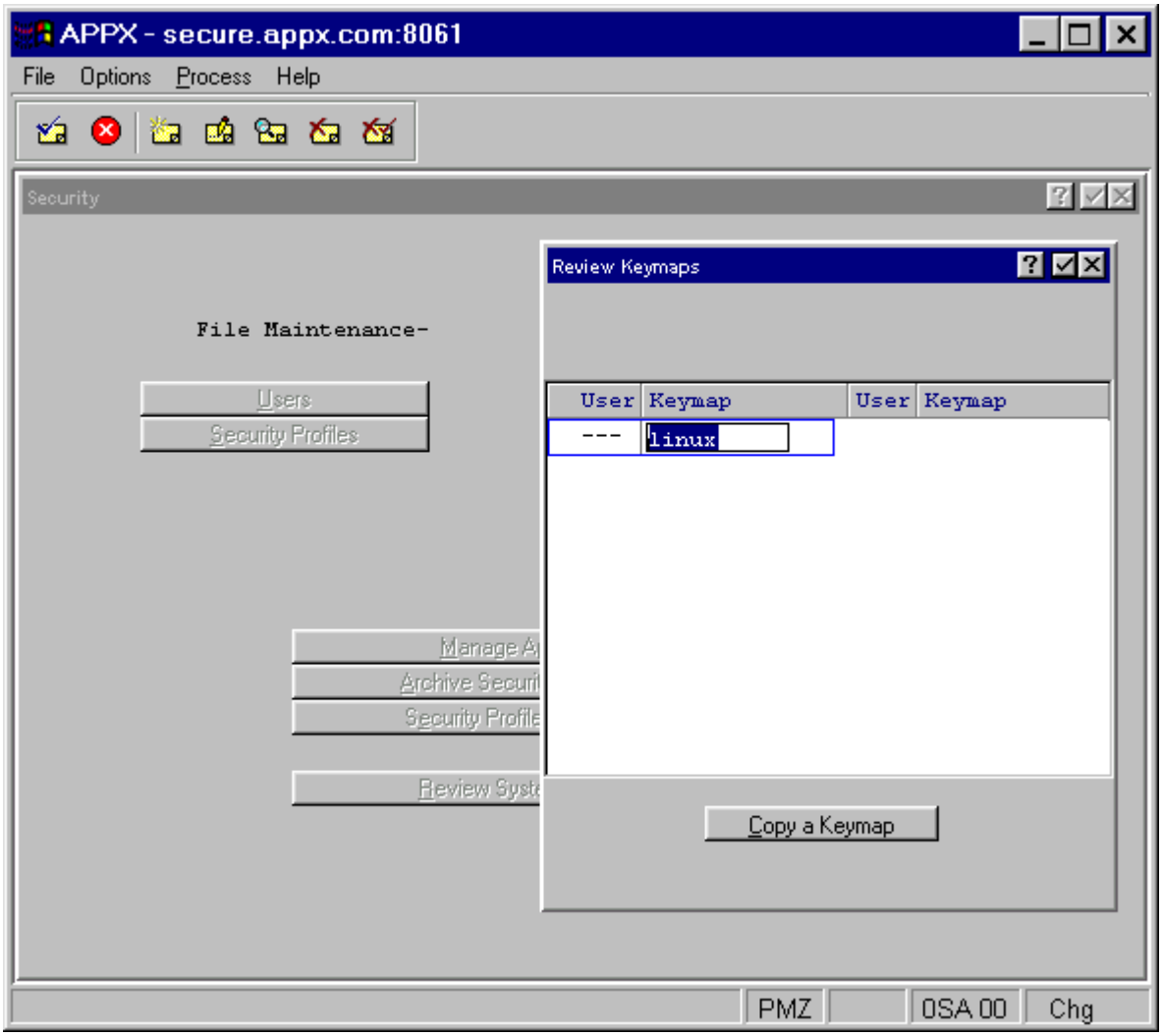


Figure 2-58. Review System Default Keymaps Overlay

Language Translation

The LANGUAGE TRANSLATION option on the System Administration Menu invokes the Language Translation Menu shown in *Figure 2-59*. The options on this menu are used to control what language or languages APPX applications are presented to users in. For instructions on using this feature, please refer to the separate *Language Translation* manual available from APPX or your APPX dealer.

Language Translation menu

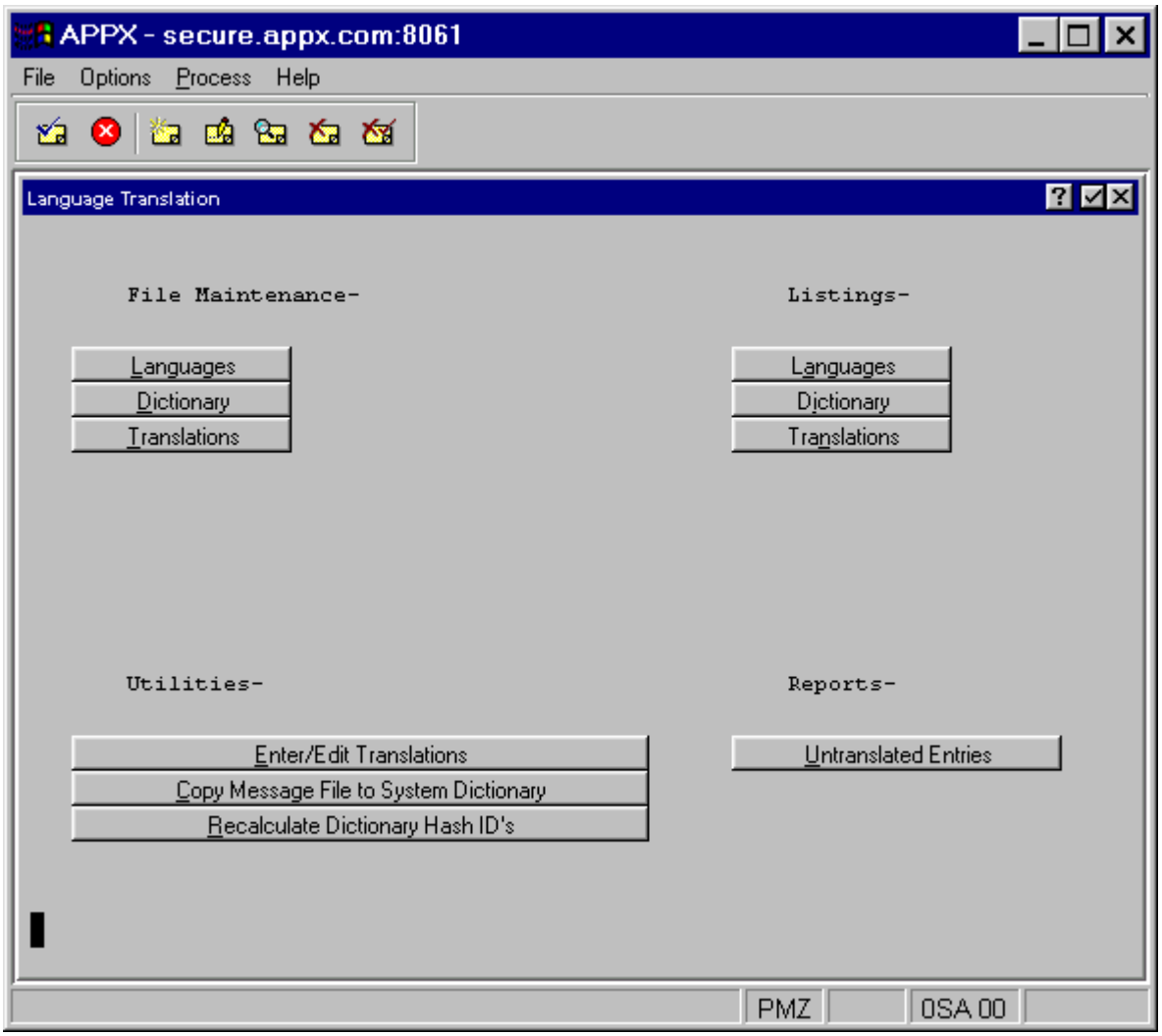


Figure 2-59

CHAPTER 3

Application Database Management

Overview - Database Management

APPX provides the capability for you to manage the data files associated with the applications at your installation. When a new application is developed or installed, you need to specify certain file creation attributes, such as number of records and file location, and create all of the files that are to be included in the application. You may also need to restructure one or more files if changes to the data dictionary require a structural change (such as increasing a field size in a record).

There are three types of files within the APPX environment:

- ***System administration files***, described in System Administration File Management under System Setup in Chapter 2, allow you to define your APPX installation.
- ***Design files*** contain the specifications that define each application running or under development at your installation.
- ***Data files*** contain the records that are associated with each application, such as payroll or customer data.

APPX provides the facilities to manage all three types of files. The options provided for each type are very similar with one exception: APPX allows you to restructure data files, and possibly design files, while the format of system administration files is fixed. The *Restructure* option does not appear on the latter menu. The option of restructuring design files should only be used if your upgrade or installation instructions specifically request it. Otherwise, the instructions provided in the following sections apply to all three aspects of database administration.

Database Management Menu

When you select DATABASE MANAGEMENT from the APPX main menu, you must identify both the database and the application in order to perform database management functions. The Database Management menu is shown in *Figure 3-1* below.

Database Management menu

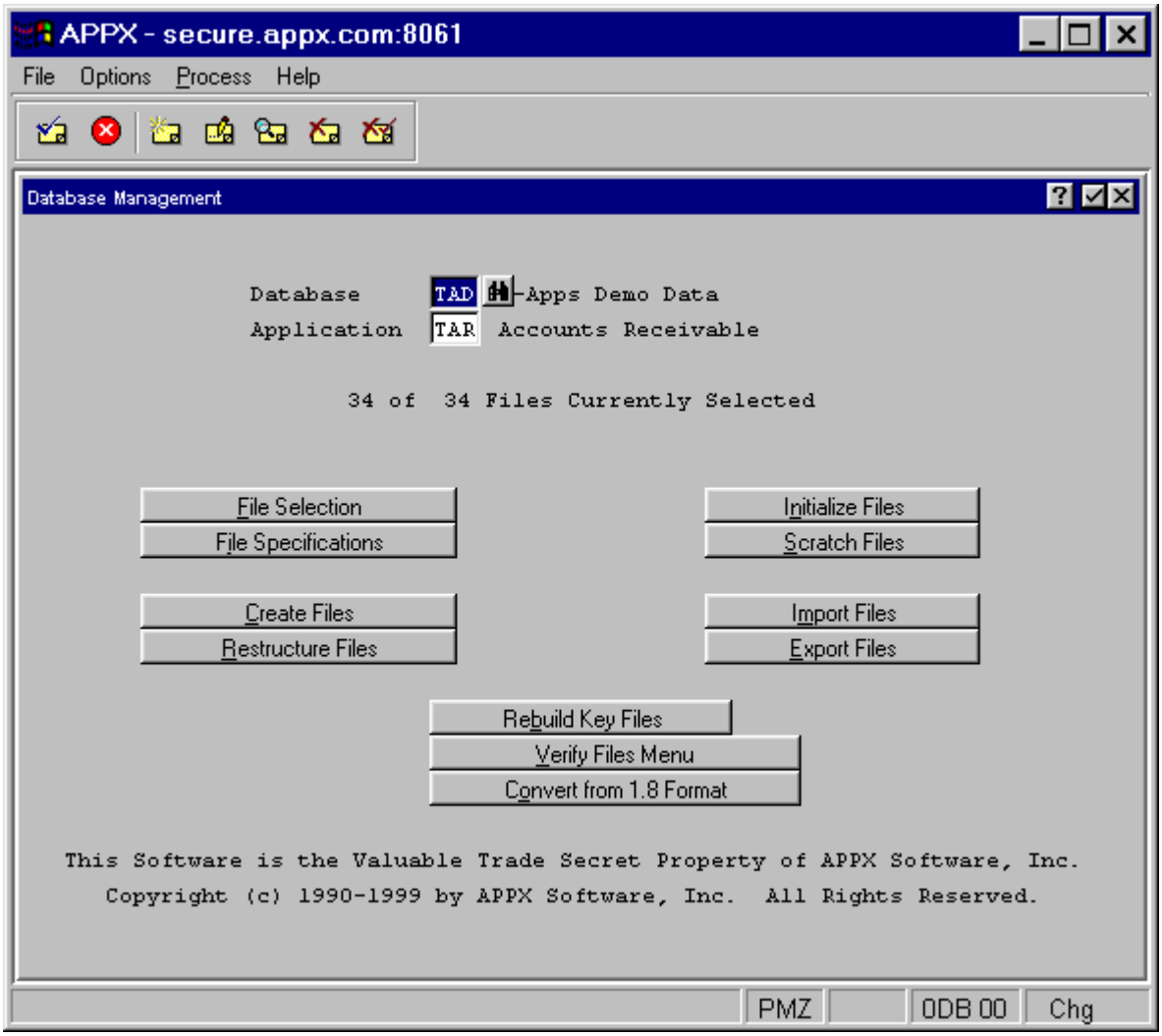


Figure 3-1

File Selection

Identifies which files you want to work on. See the File Selection section for details.

File Specifications

Provides information about the files selected previously with the FILE SELECTION option See the File Specifications section for details.

Create Files

Creates the files selected previously with the FILE SELECTION option (assuming they do not already exist) unless the estimated number of records is zero. Once completed, you do not normally have to run this process again. As each file is addressed, its file name appears on the screen along with the status of creating. If there are any problems encountered, APPX displays an appropriate message.

Restructure Files

Restructures the data files selected previously with the FILE SELECTION option. You use this option when, for example, you add a new field to a record or expand the length of an existing field. As each file is addressed, its file name appears on the screen along with the status of restructuring. If there are any problems encountered, APPX displays an appropriate message.

Initialize Files

Sets up files with no records by scratching and recreating the files selected previously with the FILE SELECTION option. Before processing begins, a warning message appears because the initialization process destroys data. As each file is addressed, its file name appears on the screen along with the status of initializing. If there are any problems encountered, APPX displays an appropriate message.

Scratch Files

Deletes the data files selected previously with the FILE SELECTION option and eliminates them from the system directory. Before processing begins, a warning message appears because the scratch process destroys data. As each file is addressed, its file name appears on the screen along with the status of scratching. If there are any problems encountered, APPX displays an appropriate message.

Import Files

Reads in files that were generated on another computer system using the EXPORT FILES option. You normally only use this option when you are converting from one hardware platform to another. See the Import Files overlay for details.

Export Files

Generates standard, fixed-length ASCII records in a portable format for the files selected previously with the FILE SELECTION option. You normally only use this option when you are converting files from one hardware platform to another.

Rebuild Key Files

Repairs corrupted APPXIO index files for files with FMS type 1 only which were selected previously with the FILE SELECTION option. See the Rebuild Key Files section for details and for ways to address files with FMS type other than 1.

Verify Files Menu

Contains utilities used to verify the structure and integrity of APPX files selected previously with the FILE SELECTION option. See the Verify Files Menu overlay for details.

Convert from 1.8 Format

Converts the files selected previously with the FILE SELECTION option from pre-version 2.0 format to the 2.0 format. Since the older versions of APPX used a file structure which is incompatible with the newer versions, this utility must be run when upgrading in order to restructure files. Any errors APPX encounters during this process are displayed on your screen and you also have the option of printing an error log upon completion.

File Selection

Normally, you work on all of an application's files at the same time when you first set them up. You may, however, have occasion to work on only one or selected files. The File Selection option from the Database Management menu allows you to specify which files you want to work on. The File Selection screen is shown in *Figure 3-2* below:

File Selection screen

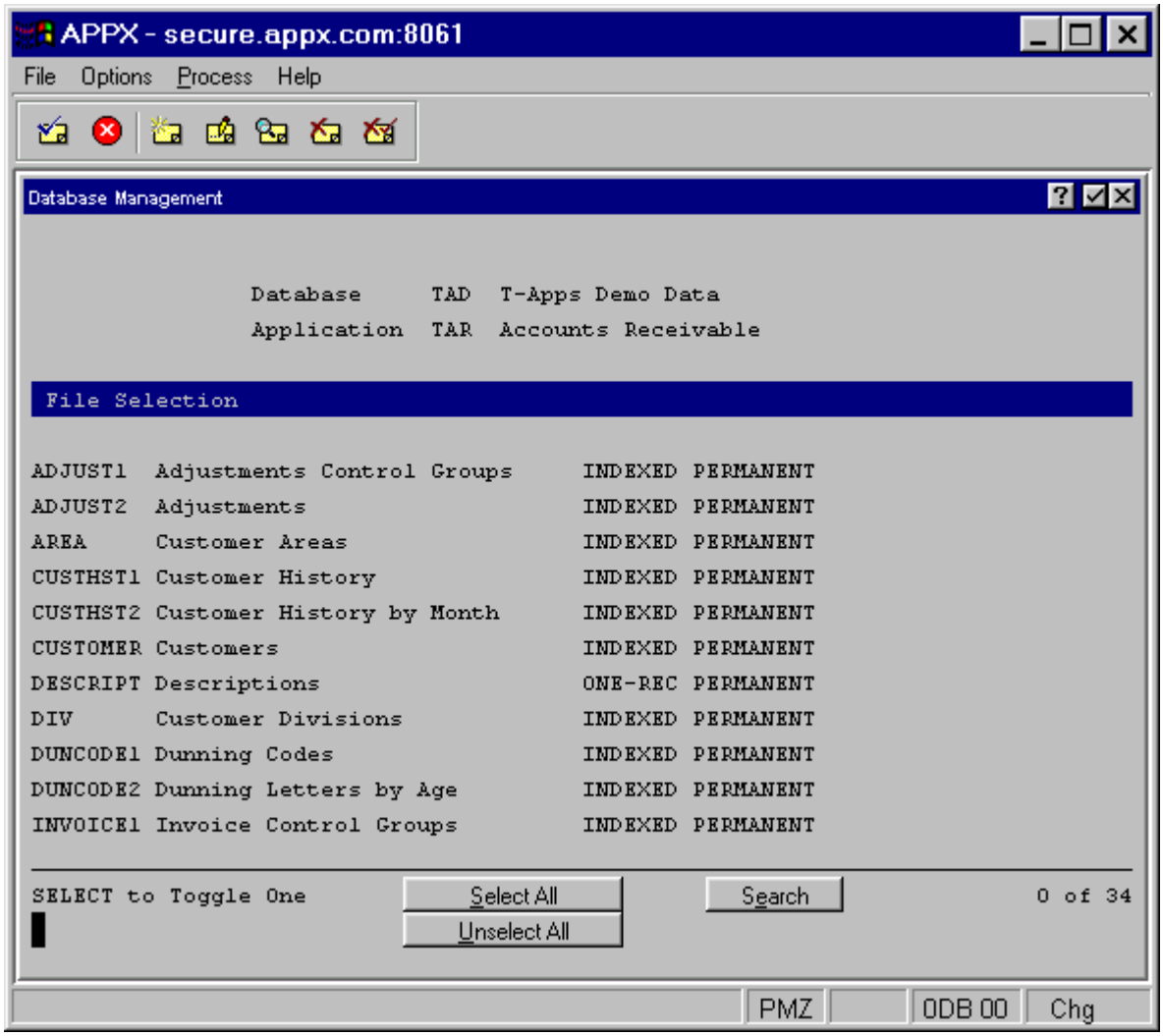


Figure 3-2

The File Selection screen contains two columns that list the names and corresponding descriptions of all the files that are specified in the data dictionary for the application. The following options allow you to designate the files you want:

- `PAGE DOWN` displays the next screen of files. To end file selection, select `END`.

Select to Toggle One

Allows you to "toggle" between selecting and unselecting one file at a time. If a file is selected, the name and description become bright (highlighted). Using the arrow keys, or the `PAGE UP` or `PAGE DOWN` keys, move the cursor onto any character of the desired file name. Then press the SELECT key (F3) or click `ENTER` in the title bar to select or unselect files one at a time.

Select All

Highlights (selects) all of the files for further use.

Unselect All

Unhighlights (unselects) all of the files. If you end with no files selected, subsequent file-related options have no effect, since there are no files designated.

Search

Allows you to search for a particular file.

File Specifications

You use the FILE SPECIFICATIONS option on the Database Management menu to provide information about each of the files that you intend to create. The first screen is a standard scrolling display. You can use all of the scrolling options as described in the APPX *User Manual*. The File Specifications screen is shown in *Figure 3-3* below.

File Specifications screen

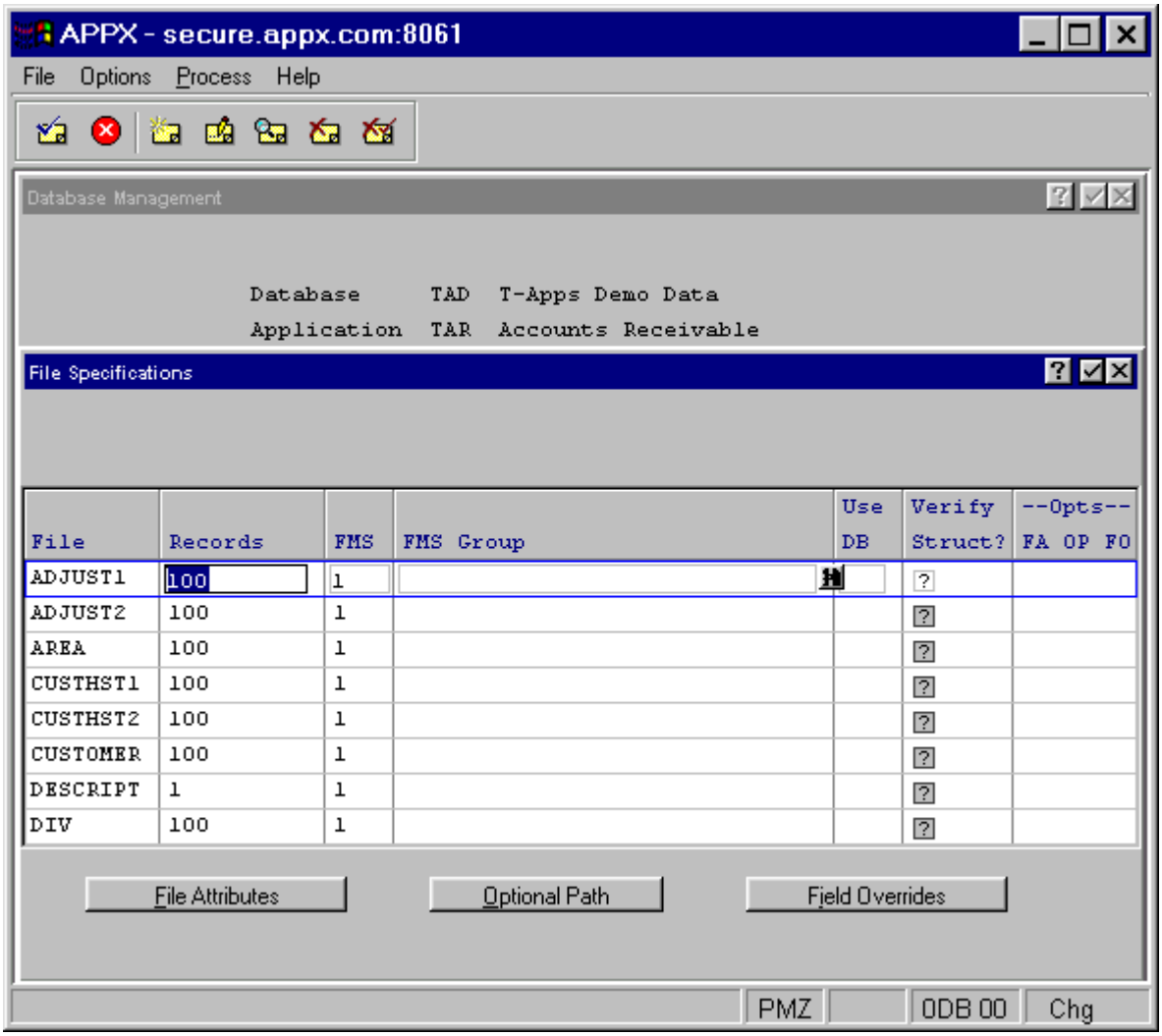


Figure 3-3

The following is a description of each field on the File Specifications screen:

- ***File Name*** (not editable) displays the file name.

Records

Indicates an estimate for the number of records the file will contain. Note that this record count is not used in actually sizing the file (except on the DEC/VAX which is no longer supported). Instead, APPX will create the file as small as possible and then allocate disk space dynamically as records are added. However, if you enter a value of *0* (zero), APPX will not create the file when you select the CREATE FILES option.

- ***FMS*** (file management system) is a system-dependent value that designates a specific file management system on your computer. See File System Groups File Maintenance for a description of FMS values.

FMS Group

Is the name of the file management group that this file belongs to. FMS Group and FMS combined reference the File System Groups file in System Administration, which provides default values for certain file specifications. Click on the *Scan* icon to choose an appropriate FMS Group for each file.

- ***Use DB*** (use database) allows multiple databases to share a single file or set of files. If you enter a value in this field and the estimated number of records you specified for the file is zero, APPX uses the database that you specify here when referencing the file.

Verify Struct?

(Verify structure?) Select "no" ☐ if you want APPX to skip verification of the structure of the file each time a user references it. Select "yes" ☑ if you want verification to occur. If you select "no" and the structure of any of the files changes, users may receive an error message at run time. During application development, you may want to verify structures routinely until the structures stabilize.

These options are also available on the File Specifications screen:

File Attributes

Calls an overlay for entering control parameters that may be needed for external file systems. The specific overlay called varies by FMS type (see File Attributes overlays for ISAM, Sybase, Oracle, ODBC, and APPX/Net). These overlays mirror the ones for FMS Controls for ISAM, Sybase, Oracle, ODBC, and APPX/Net found within File System Groups File Maintenance under Configuration in System Administration. File Attributes entries differ in that they apply only to the specified file and override any entries in FMS Controls. For example, you can specify the actual external *Table Name* for a given file. Fields left blank on these screens will assume the values entered for the corresponding file system group.

Optional Path

Calls the Optional Path Name overlay which allows you to override the standard default path name that APPX automatically assigns. This is a system-specific value. Refer to your APPX installation guide for information on how to assign values. If this field is left blank, APPX uses the combined *FMS Group* and *FMS* values that you specified above to retrieve the FMS path name from the File System Groups file. If the *FMS Group* on the File Specifications screen is blank, APPX uses standard values.

Field Overrides

Allows you to specify certain field characteristics for fields in external databases (such as Oracle, Sybase, ODBC, etc.). When selecting this option, a list of fields in the file appears. To set the overrides for a specific field, double click the desired field to select it and display the Field Overrides Detail overlay.

File Attributes Overlay for ISAM

If you select the FILE ATTRIBUTES option in File Specifications for a file with FMS type 1, 2, or 3, the File Attributes overlay for ISAM appears as shown in *Figure 3-4* below:

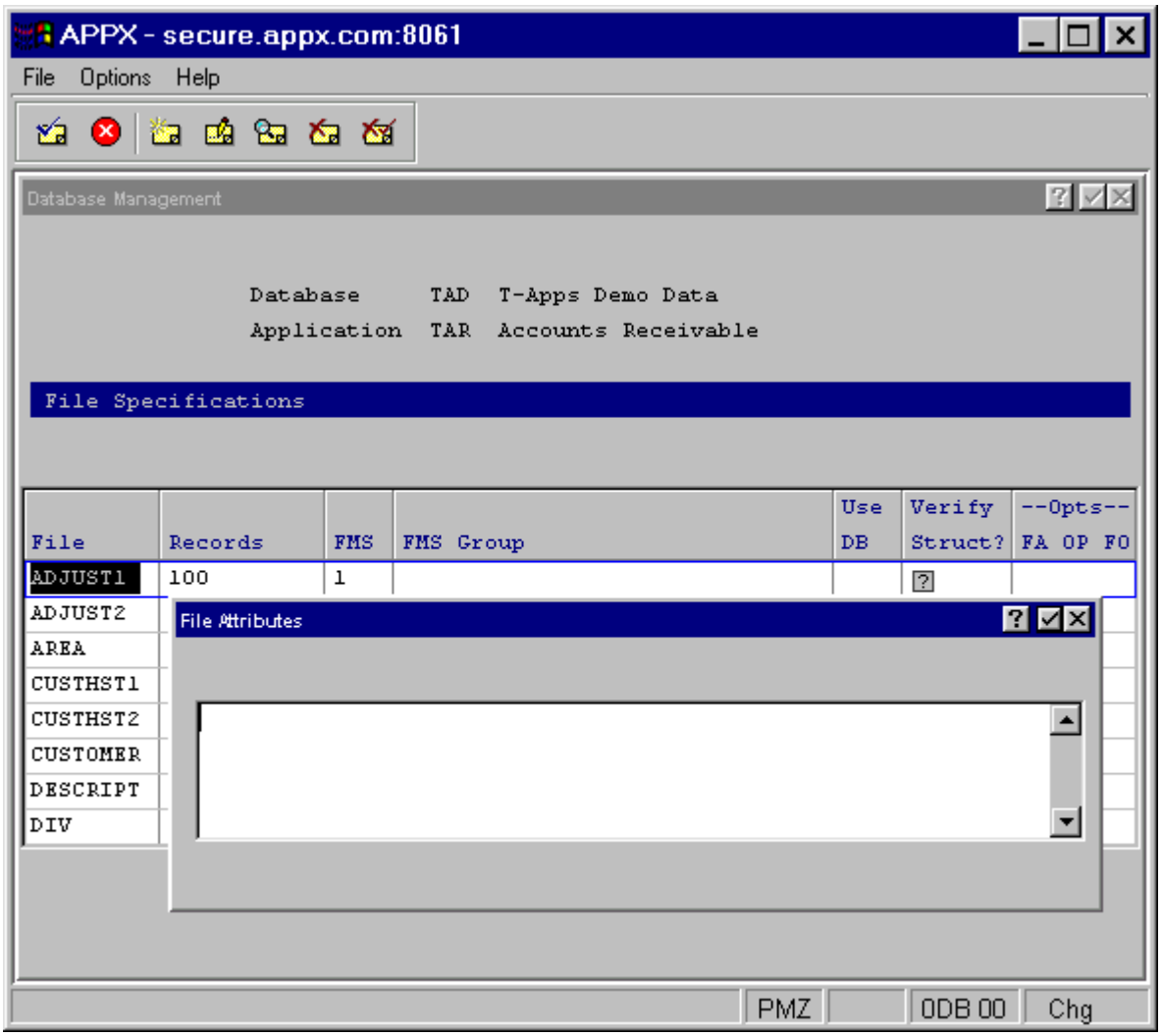


Figure 3-4. File Attributes Overlay for ISAM (Types 1,2,3)

File Attributes Overlay for Sybase

If you select the FILE ATTRIBUTES option in File Specifications for a file with FMS type 4, the File Attributes overlay for Sybase appears as shown in *Figure 3-5* below:

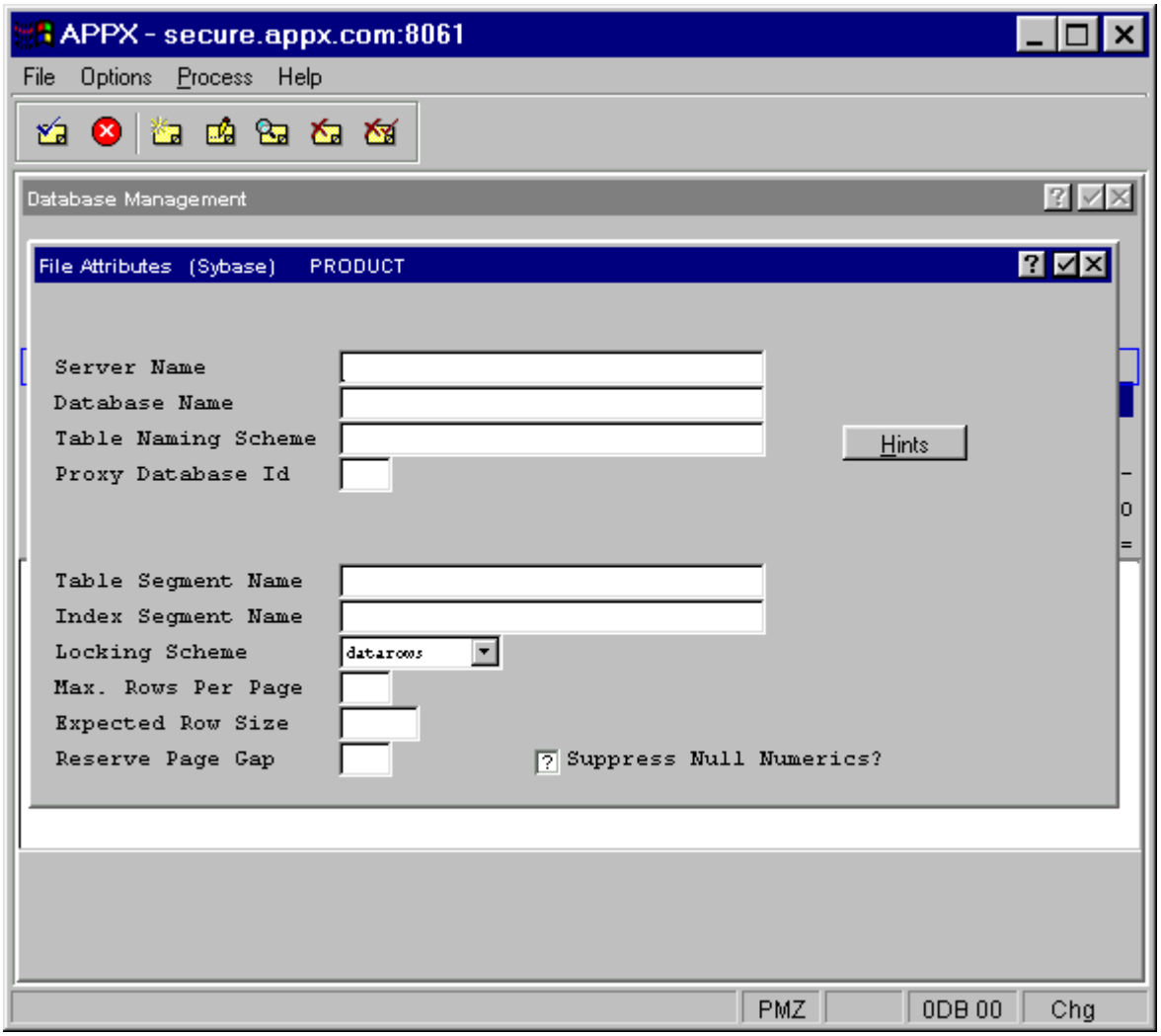


Figure 3-5. File Attributes Overlay for Sybase (Type 4)

Entries made on this overlay apply only to the selected file and override any entries made during File System Groups File Maintenance. Fields left blank will assume the values entered for the file system group to which this file belongs. For detail field descriptions, see the FMS Controls Overlay for Sybase. However, note that the field labeled *Table Naming Scheme* can contain either a scheme such as those found on the Hints overlay or the table name itself for this specific file.

Select the HINTS option for help on constructing a table scheme or name. See the Table Naming Hints section for more information.

File Attributes Overlay for Oracle

If you select the FILE ATTRIBUTES option in File Specifications for a file with FMS type 5, the File Attributes overlay for Oracle appears as shown in *Figure 3-6* below:

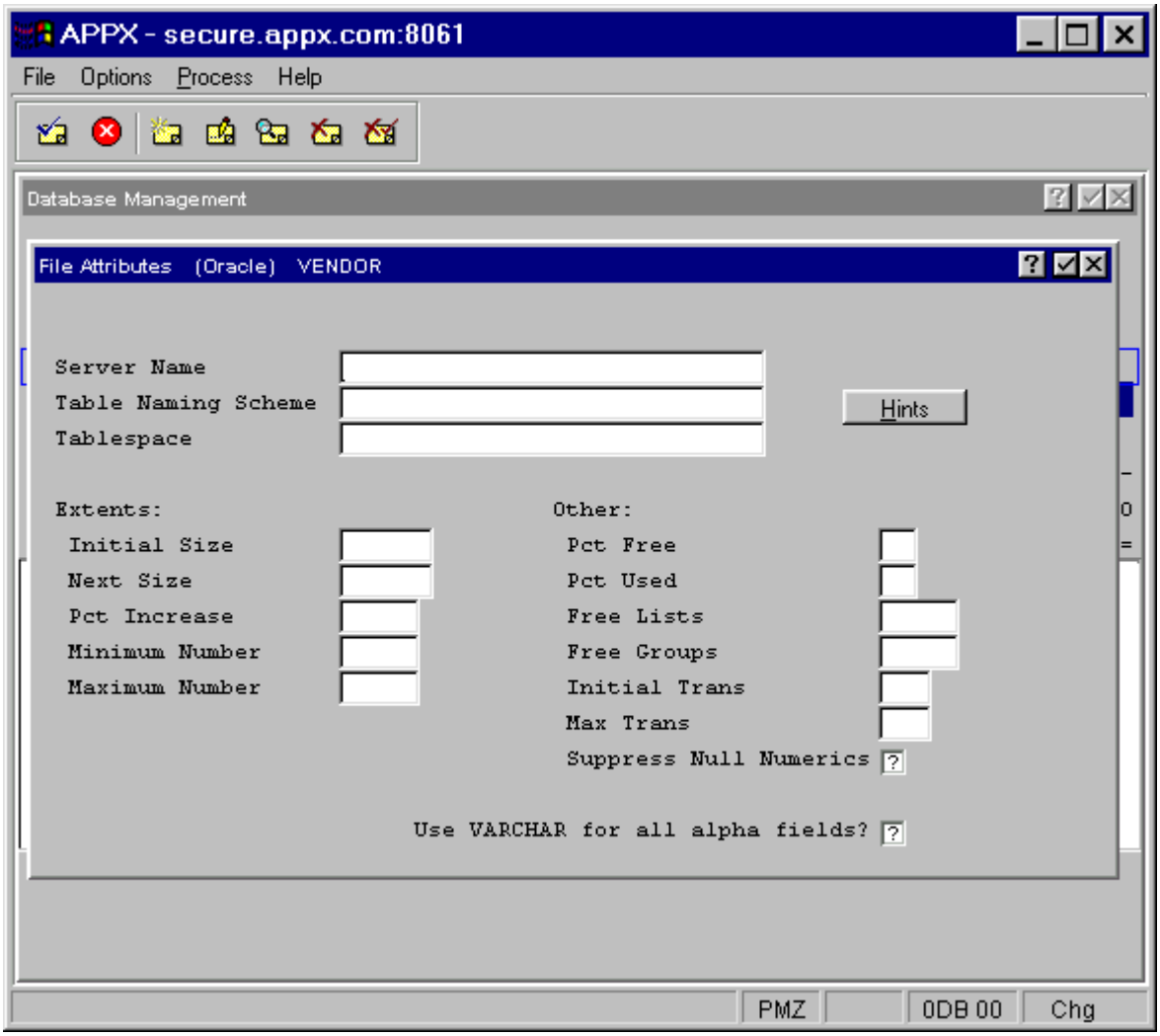


Figure 3-6. File Attributes Overlay for Oracle (Type 5)

Entries made on this overlay apply only to the selected file and override any entries made during File System Groups File Maintenance. Fields left blank will assume the values entered for the file system group to which this file belongs. For detail field descriptions, see the FMS Controls Overlay for Oracle. However, note that the field labeled *Table Naming Scheme* can contain either a scheme such as those found on the Hints overlay or the table name itself for this specific file.

Select the HINTS option for help on constructing a table scheme or name. See the Table Naming Hints section for more information.

File Attributes Overlay for ODBC

If you select the FILE ATTRIBUTES option in File Specifications for a file with FMS type 6, the File Attributes overlay for ODBC appears as shown in *Figure 3-7* below:

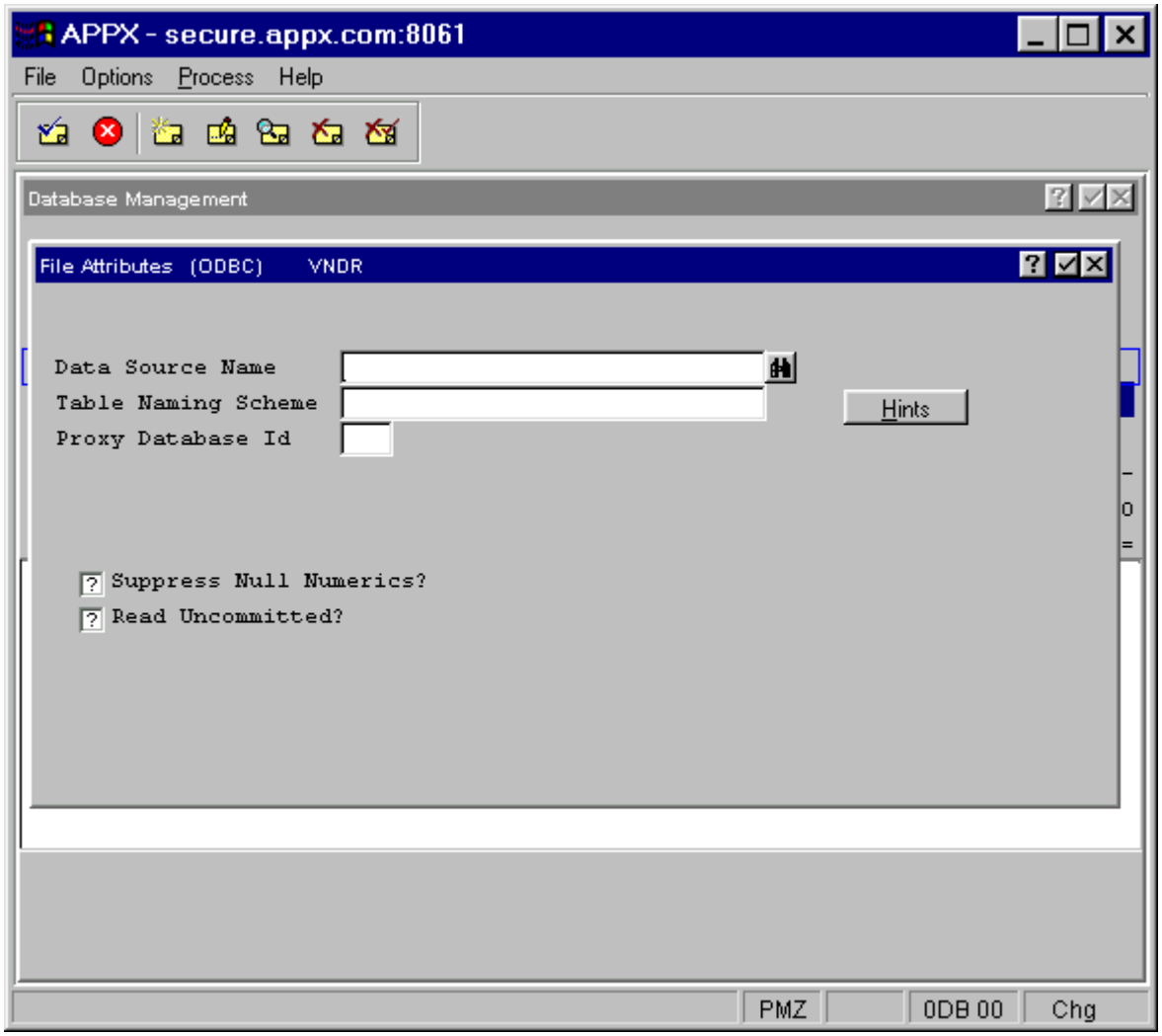


Figure 3-7. File Attributes Overlay for ODBC (Type 6)

Entries made on this overlay apply only to the selected file and override any entries made during File System Groups File Maintenance. Fields left blank will assume the values entered for the file system group to which this file belongs. For detail field descriptions, see the FMS Controls Overlay for ODBC. However, note that the field labeled *Table Naming Scheme* can contain either a scheme such as those found on the Hints overlay or the table name itself for this specific file.

Select the HINTS option for help on constructing a table scheme or name. See the Table Naming Hints section for more information.

> **Note:** APPX does not support fields (or records) longer than 32,767 (32K) characters. If you map an APPX file to a pre-existing RDBMS table, you may find that the table includes a column larger than 32K; such columns are often referred to as *memo* columns. APPX offers limited support for memo columns. You can access a memo column by defining (in the APPX data dictionary) an ALPHA X(***n***) field in the corresponding position.

When you read a record that includes a memo column, APPX will only read the first ***n*** characters and will truncate any remaining data. If you read a record *with hold* and APPX would have to truncate a memo field, the read will fail. Reading a record *with hold* implies that you may try to update the record and APPX will try to prevent you from losing data through truncation. If you need to delete a record that would include truncated data, use *partial-record I/O* to avoid truncation. You can also use *partial-record I/O* to update other fields in the record. In general, you should avoid reading memo columns except in those cases where you actually need the data.

Table Naming Hints

If you select the HINTS option from the File Attributes overlays for Sybase, Oracle, and ODBC, the Table Naming Hints overlay appears as shown in *Figure 3-7-A* below:

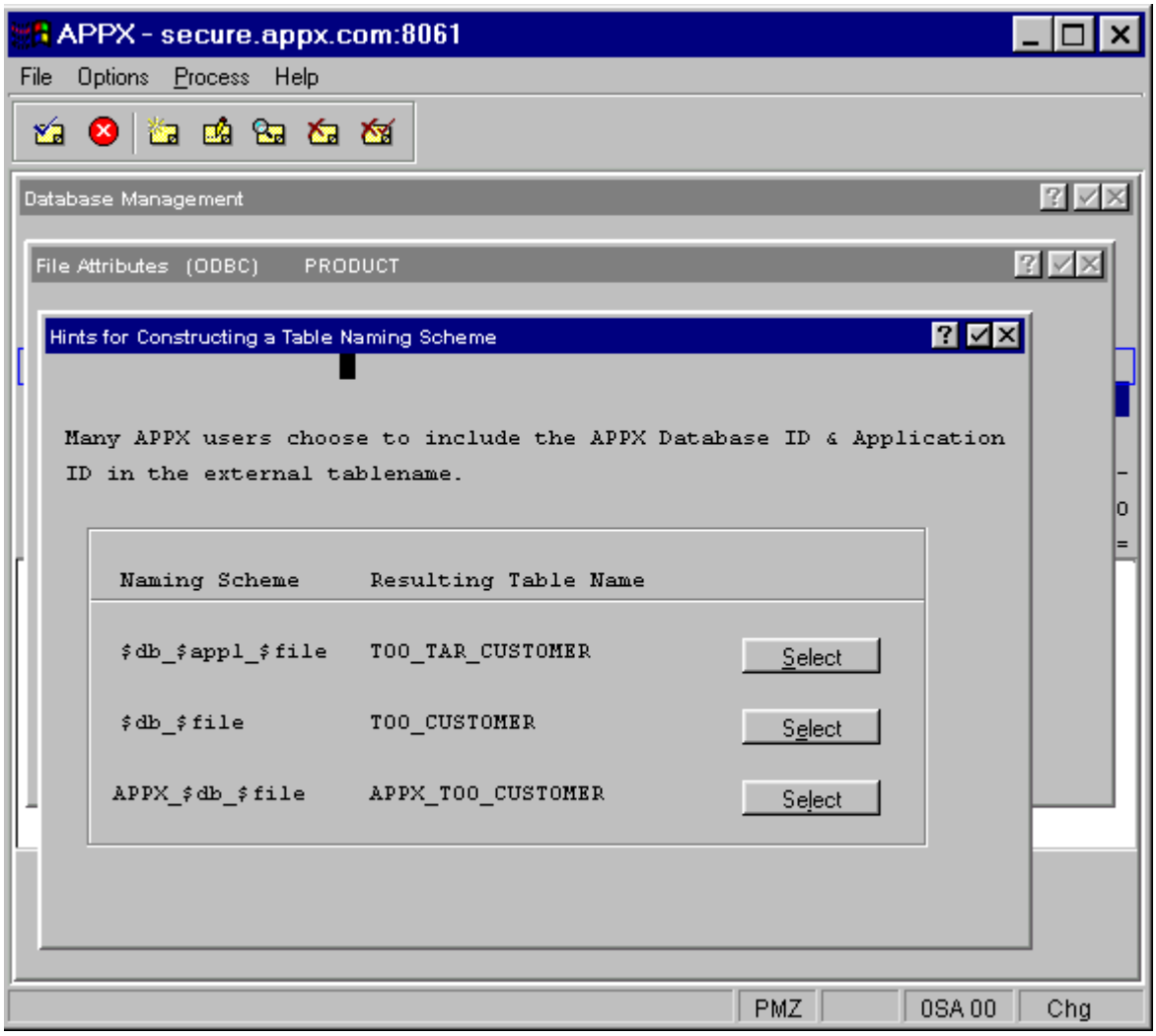


Figure 3-7-A. Table Naming Hints Overlay

The Hints overlay contains this option by each naming scheme example:

- ***Select*** defaults the associated naming scheme into the File Attributes overlay.

There are other macros available for use in a table naming scheme besides the examples displayed on the screen. A complete list of macros is provided below:

- $host host name
- $engine APPX engine version
- $endian 'intel' or 'risc'
- $root $APPXPATH
- $uid 3-character APPX User ID
- $user OS User ID
- $pid OS Process ID
- $appl File's application ID
- $ver File's application Version
- $db File's Database ID
- $scope Process nesting level
- $file 8-character file name
- $type "Data", "Struct", "Portdata", etc.
- $anything-else An environment variable

For example, the normal location for an APPX data file would be:

$root/$db/$ap/$type/$file

If you wanted to invert the database/application hierarchy, it would be:

$root/$ap/$db/$type/$file

File Attributes Overlay for APPX/Net

If you select the FILE ATTRIBUTES option in File Specifications for a file with FMS type 7, the File Attributes overlay for APPX/Net appears as shown in *Figure 3-8* below:

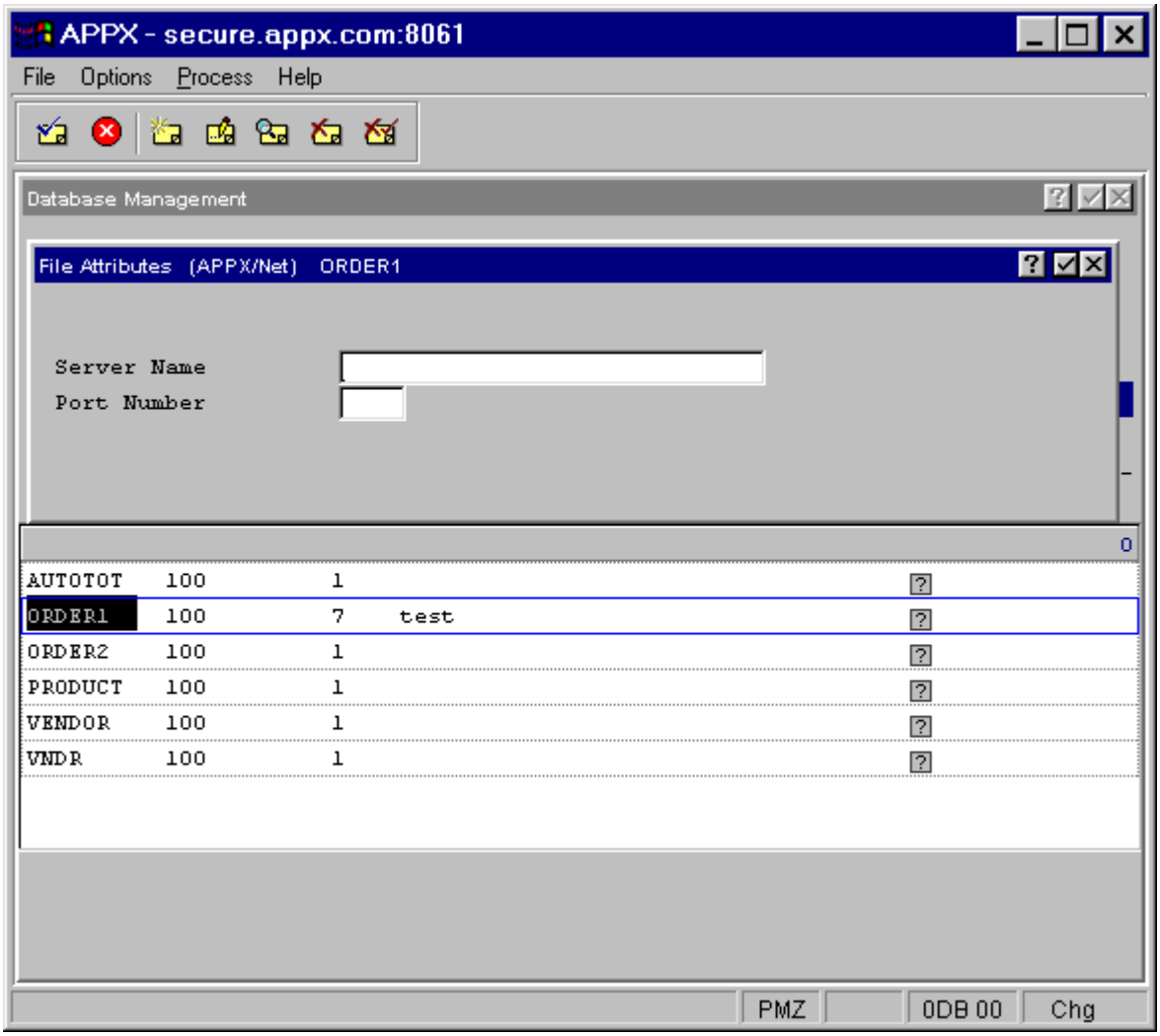


Figure 3-8. File Attributes Overlay for APPX/Net (Type 7)

Entries made on this overlay apply only to the selected file and override any entries made during File System Groups File Maintenance. Fields left blank will assume the values entered for the file system group to which this file belongs. For detail field descriptions, see the FMS Controls overlay for APPX/Net.

Optional Path Name Overlay

If you choose the OPTIONAL PATH option in File Specifications, the Optional Path Name overlay appears for the selected file as shown in *Figure 3-9* below:

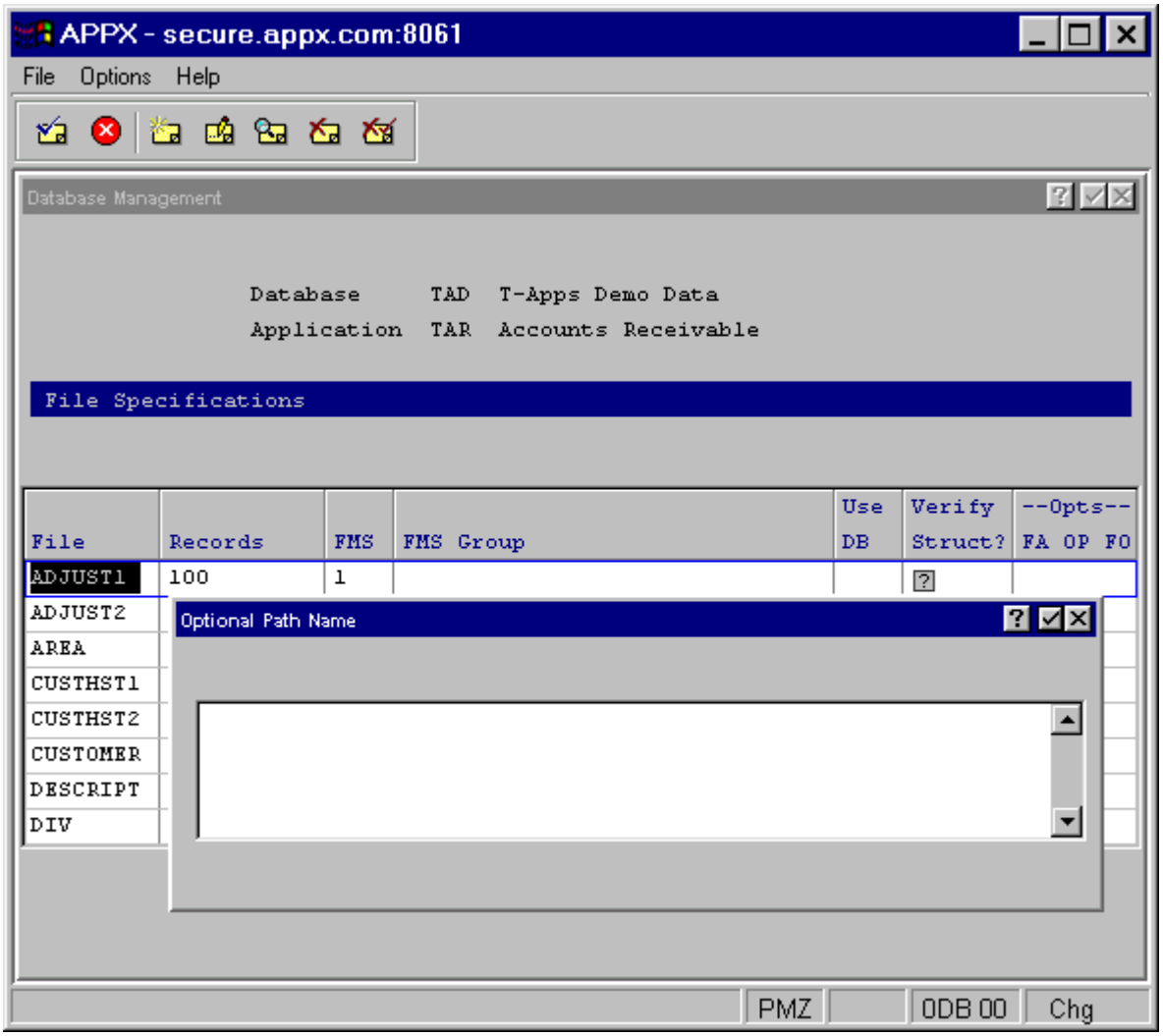


Figure 3-9. Optional Path Name Overlay

Field Overrides Detail Overlay

The FIELD OVERRIDES option in File Specifications allows you to specify certain field characteristics for fields in external databases (such as Oracle, Sybase, ODBC, etc.). Once you have chosen this option and selected the desired field, the Field Overrides Detail overlay appears as shown in *Figure 3-10* below.

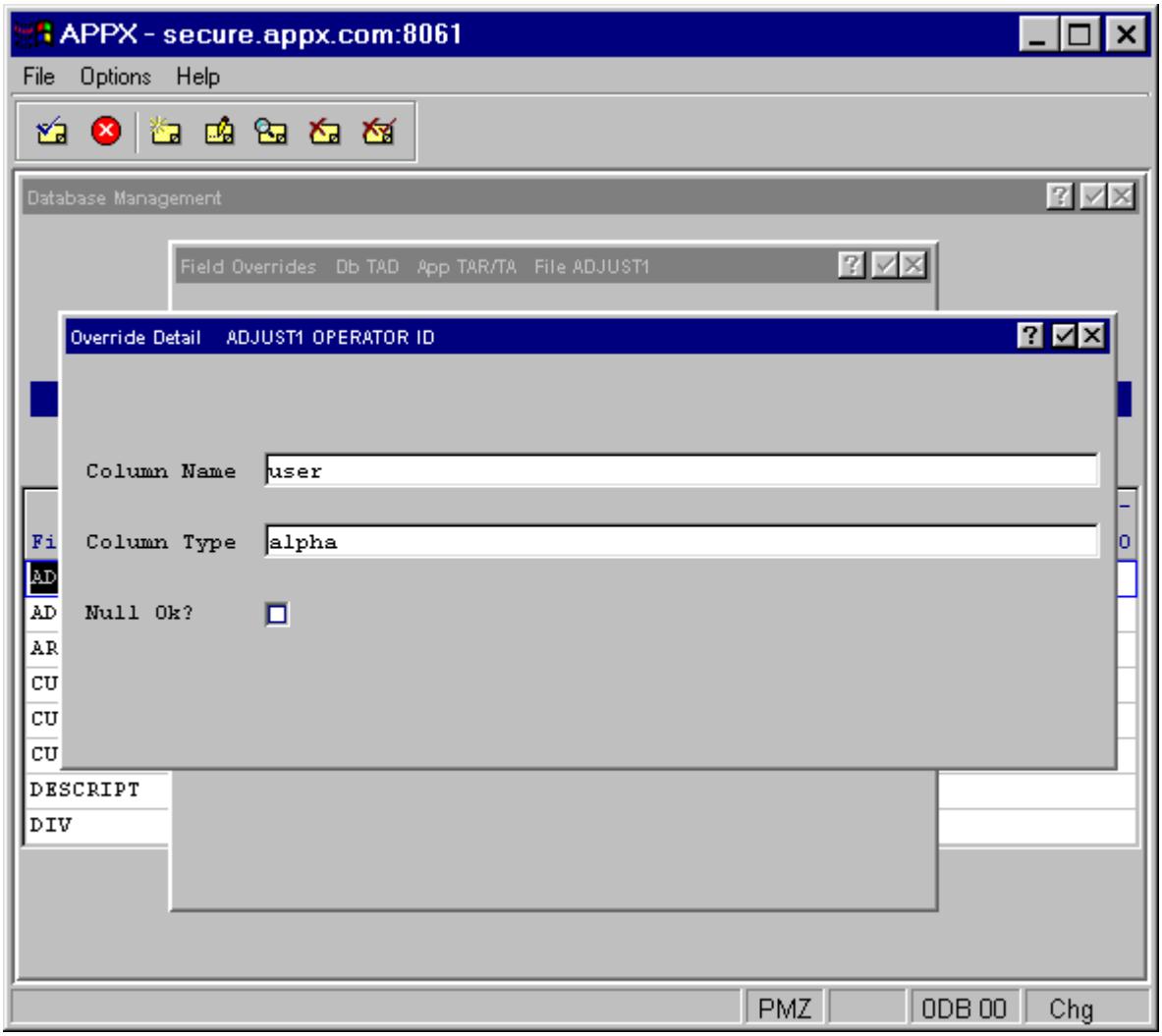


Figure 3-10. Field Overrides Detail Overlay

The following field overrides can be designated:

Column Name

Explicitly designates the column name that will be used in the external database for this field. If left blank, APPX will construct the column name from the field name.

Column Type

Explicitly designates the data type for this column in the external database. If left blank, APPX will default the data type based on the APPX data dictionary specifications for this field.

Null OK?

Indicates whether or not the value of this field can be left blank. Click the checkbox to toggle between "yes" ☑ and "no" ☐.

Import Files Overlay

If you select the IMPORT FILES option on the Database Management menu, an overlay appears as shown in *Figure 3-11* below:

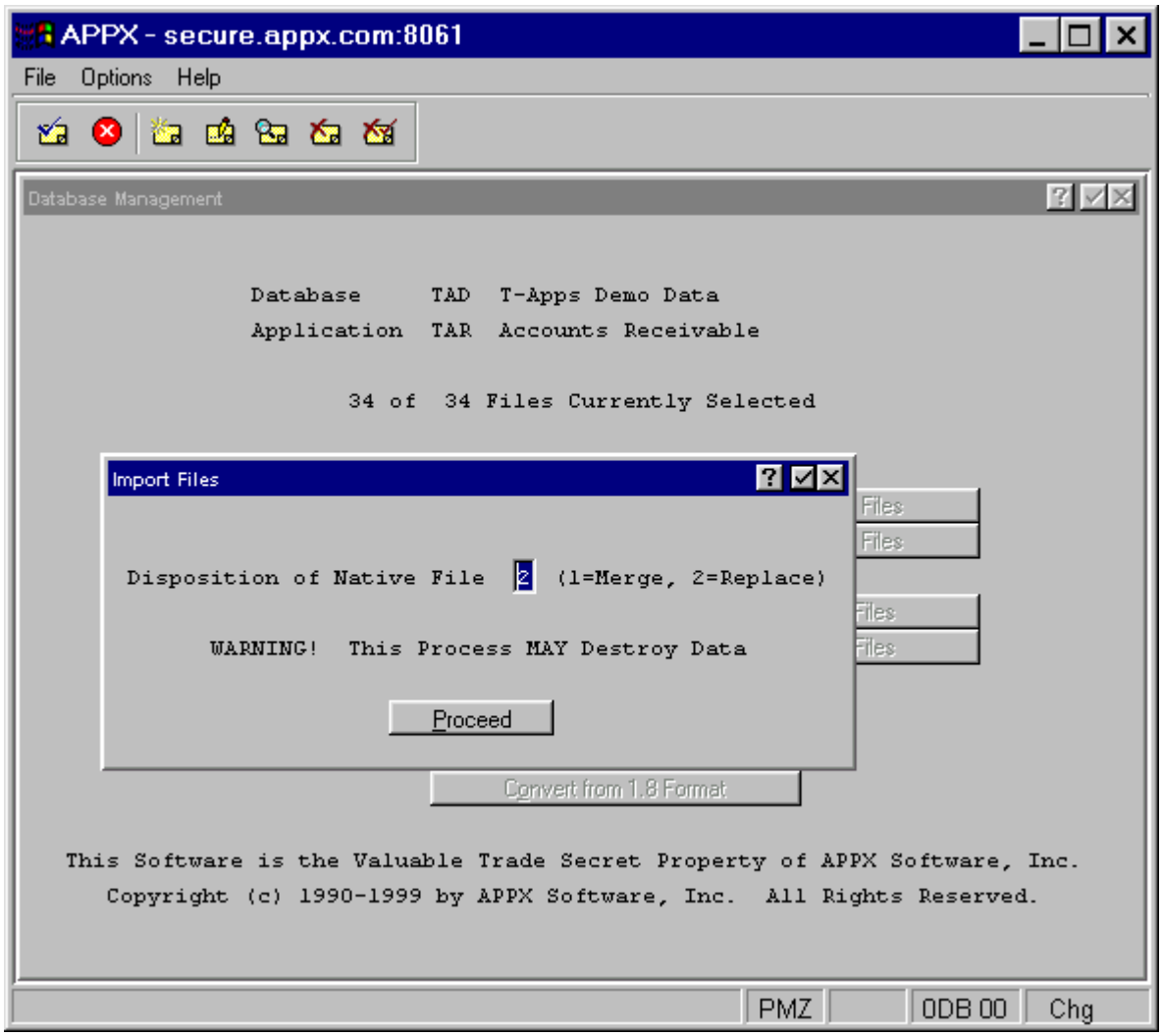


Figure 3-11. Import Files Overlay

The overlay contains a single field, *Disposition of Native File*, which determines what happens if the import file already exists on your system as follows:

- *1 = Merge* indicates that you want to merge the records from the file you are importing with an existing APPX file. Records with the same keys on each file are replaced with the record from the file that you are importing. Records that are unique to either file merge into the existing APPX file.

2 = Replace

Indicates that you want to replace any existing files with the files you are importing.

To initiate the import process, enter a *1* or *2* in the *Disposition* field as appropriate and press ENTER or click the PROCEED button. Otherwise, press END if you do not wish to proceed with the import.

Rebuild Key Files

The REBUILD KEY FILES option on the Database Management menu applies only to files with FMS type 1 (APPXIO) and can be used to repair corrupted APPXIO index files. APPXIO files consist of two physical files, the "key" file which contains all indexes for a file and the location of their corresponding data in the "data" file. This option deletes the "key" file and reads the "data" file to generate a new "key" file for those files previously selected with the FILE SELECTION option.

On Windows systems, manually delete the "key" file and run the *Create Files* option. This will create a new "key" file using existing data.

On Oracle systems, execute a DROP INDEX command using SQL*PLUS and then run the *Create Files* option. APPX will create new indexes in Oracle and leave the existing table intact.

Verify Files Menu Overlay

If you select the VERIFY FILES MENU option on the Database Management menu, an overlay appears as shown in *Figure 3-12* below. (This same overlay can also be called by the VERIFY FILES MENU option on the System Administration File Management menu.)

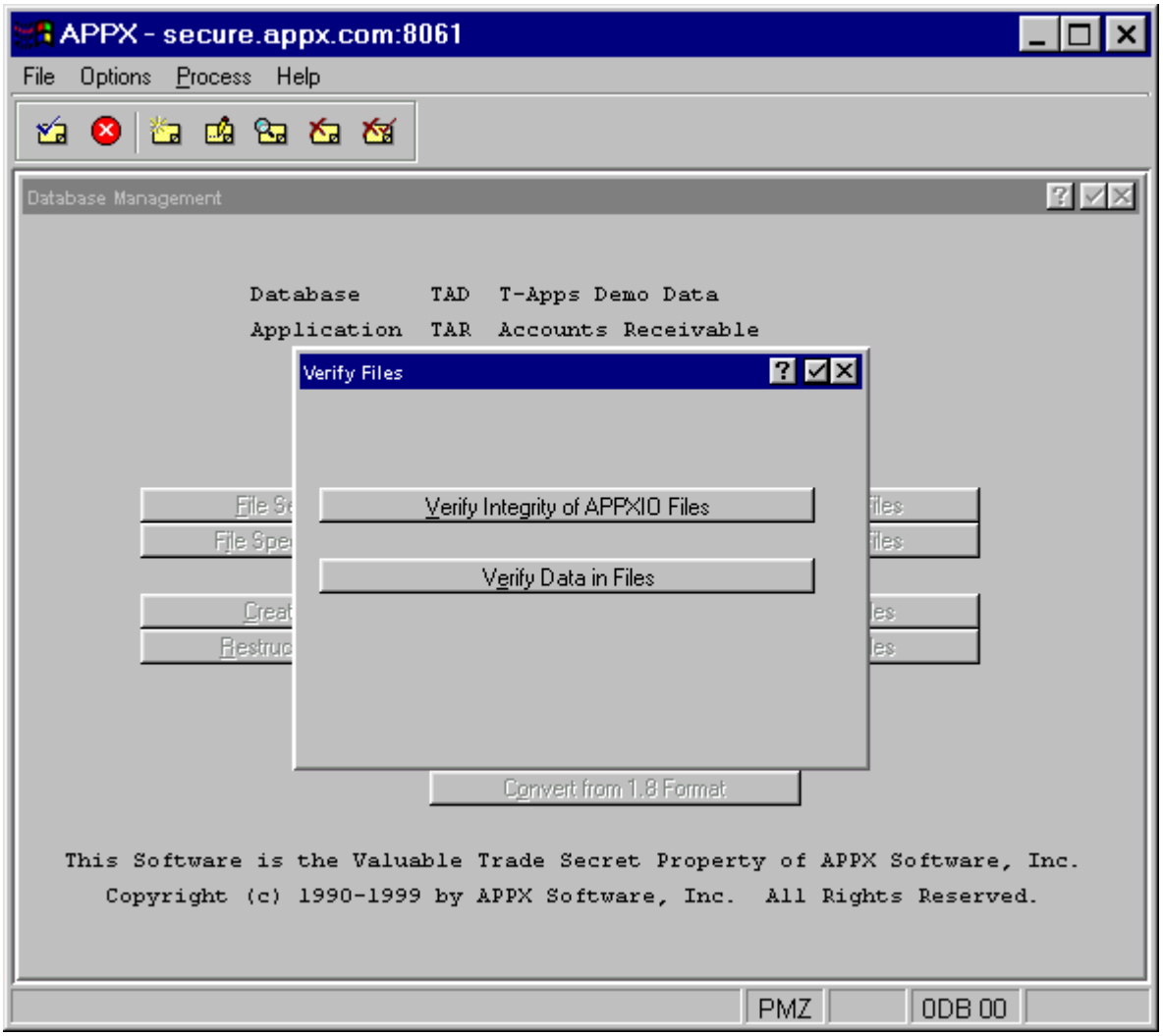


Figure 3-12. Verify Files Menu Overlay

The utilities contained on the Verify Files menu are used to verify the structure and integrity of APPX data files. These utilities consist of:

Verify Integrity of APPXIO Files

By checking that the entire data file (extension of .dat) is readable and that each record in the file contains a matching record in the index file (extension of .key). If the file passes the integrity check, the file's record count and pathname are displayed. Files that fail the check can be further analyzed by running *appxI* at the command line with a switch of "*-an*". As the title implies, this option only works on APPXIO-format files (FMS type 1).

Verify Data in Files

Checks each record in the file to ensure that the contents in each field meets the criteria of the field's *field type*. This option can be used on any type of data file. Optionally, referential integrity can be verified by checking references to *data lookups* to ensure their files and keys exist. For each file with errors, an error log is created in your default print directory. When running this utility, the following options can be selected as shown in *Figure 3-13* below:

Include Referential Integrity Check?

Causes each field with a "Data Lookup" defined to be verified against the parent data lookup file. Records with field values not in the parent file will be reported.

Report "Null" Data Lookup Values?

If referential integrity check is selected, then this flag indicates whether null (blank) values will be validated against the parent data lookup file.

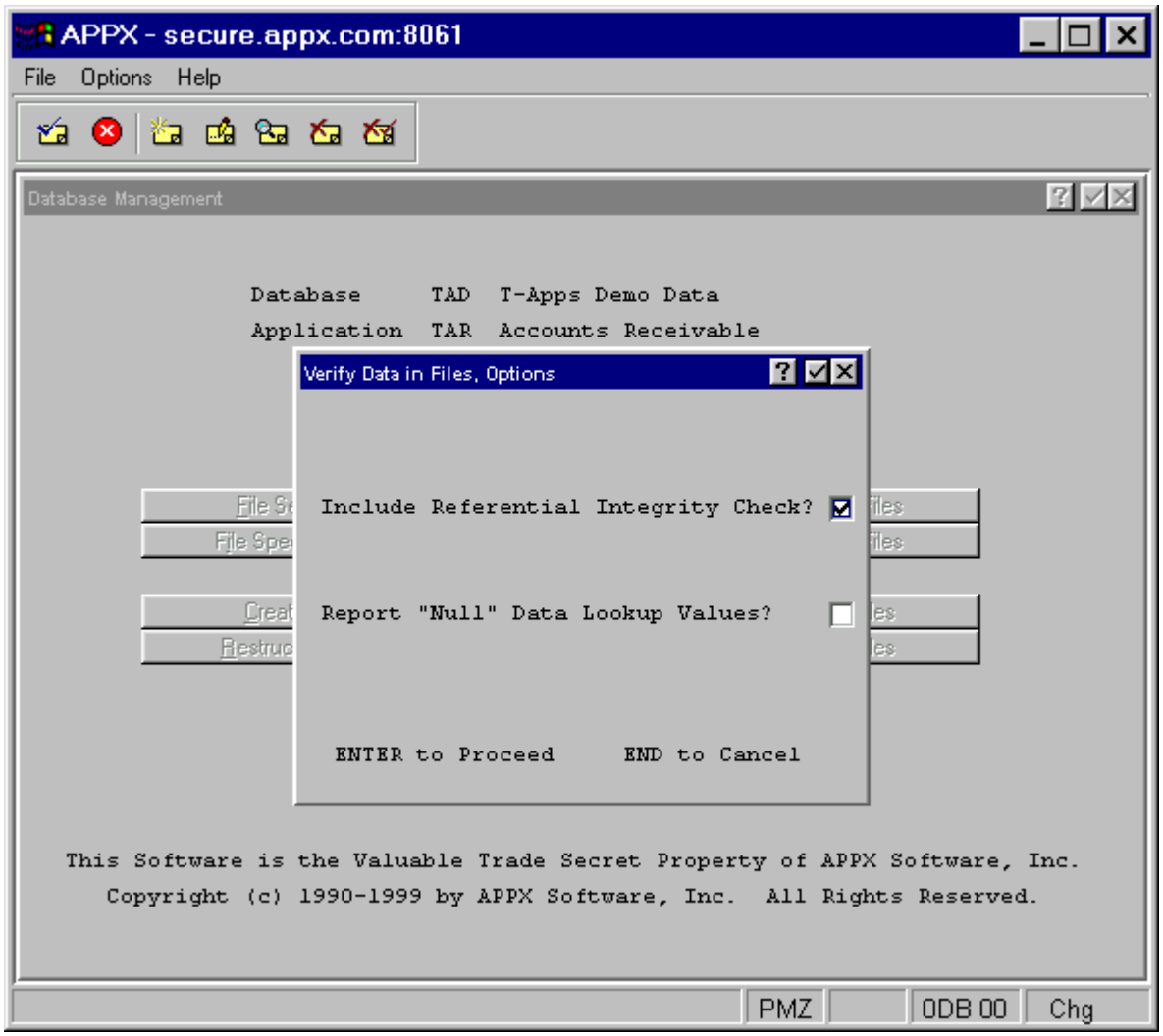


Figure 3-13. Verify Data In Files Options Overlay

CHAPTER 4

Application Documentation Management

'Manage Application Documentation' Menu

When you select DOCUMENTATION MANAGEMENT from the APPX Main Menu, the 'Manage Application Documentation' menu shown in *Figure 4-1* appears:

'Manage Application Documentation' menu

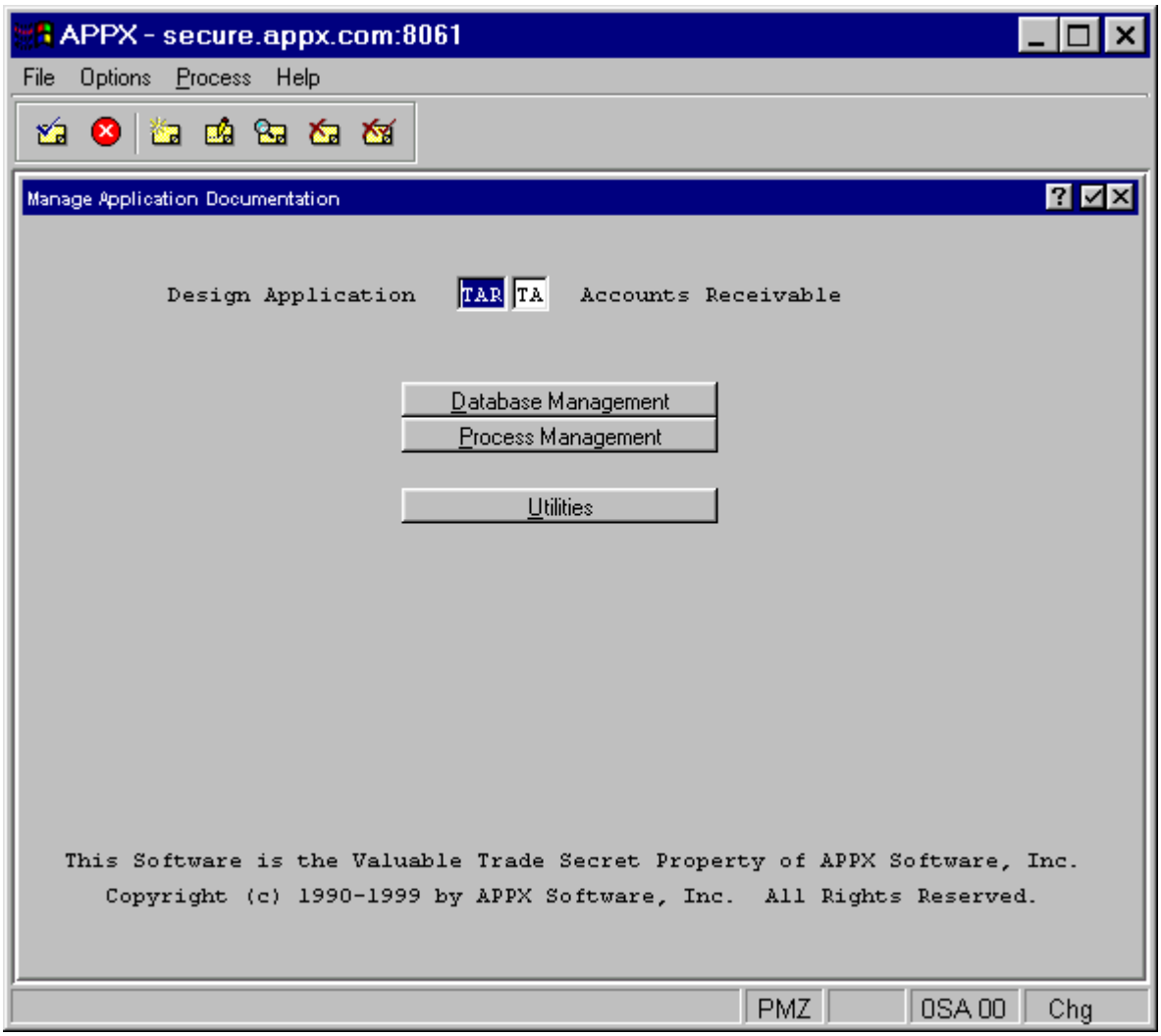


Figure 4-1

Enter the application ID and version number of the application whose documentation you wish to edit, and press ENTER. The following options are available on this menu:

- *Database Management* allows you to add, edit and delete documentation for domains, files and fields in your application.

- *Process Management* allows you to add, edit and delete documentation for all types of processes in your application.
- *Utilities* allow you to search your application for items and processes without documentation.

The sections that follow describe each of these options in detail.

Database Documentation Management

When you select the DATABASE MANAGEMENT option from the 'Manage Application Documentation' menu, the Database Documentation Management overlay appears as shown in *Figure 4-2*.

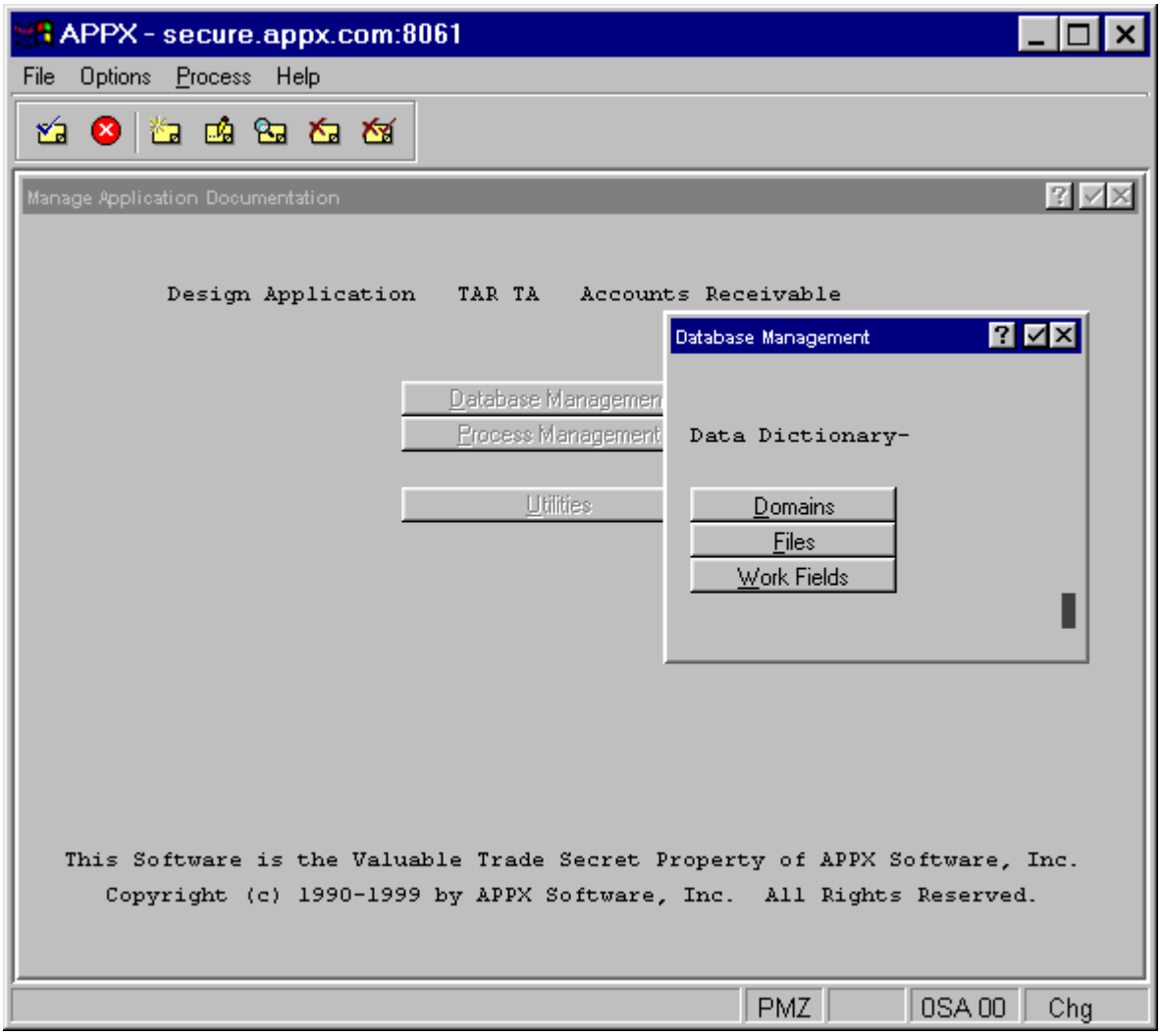


Figure 4-2. Database Documentation Management Overlay

The options on the Database Documentation Management overlay provide access to documentation for the different types of database items as follows:

- *Domains*

- *Files* (and fields)
- *Work Fields*

Choose one of these options to see the Select Item Documentation screen discussed in the next section.

Select Item Documentation Screen

When you choose the option (DOMAINS, FILES, or WORK FIELDS) for the type of item you wish to document from the Database Documentation Management overlay, a Select Item Documentation screen appears similar to the one shown in *Figure 4-3* below for files.

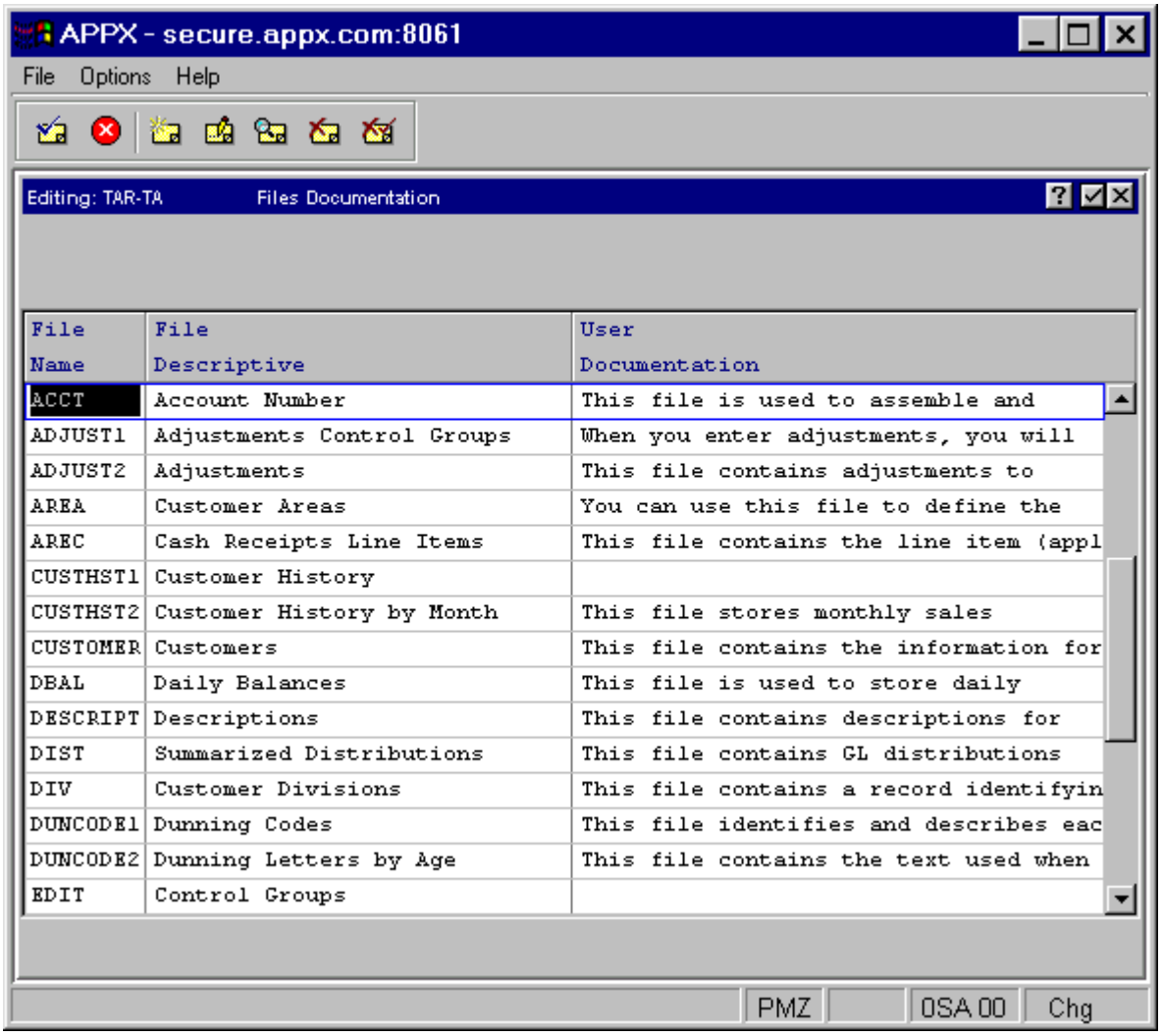


Figure 4-3. Select Item Documentation Screen (for Files)

The Select Item Documentation screen presents a scrolling display of items of the type you previously chose. Place your cursor on the specific item you wish to document and press ENTER to access the Edit Item Documentation screen discussed in the next section.

Edit Item Documentation Screen

When you choose the item you wish to document from the Select Item Documentation screen, an Edit Item Documentation screen appears similar to the one shown in *Figure 4-4* for files or in *Figure 4-5* for work fields. This screen will display any documentation that has already been entered for the item or, if no documentation has been entered, the screen will be blank.

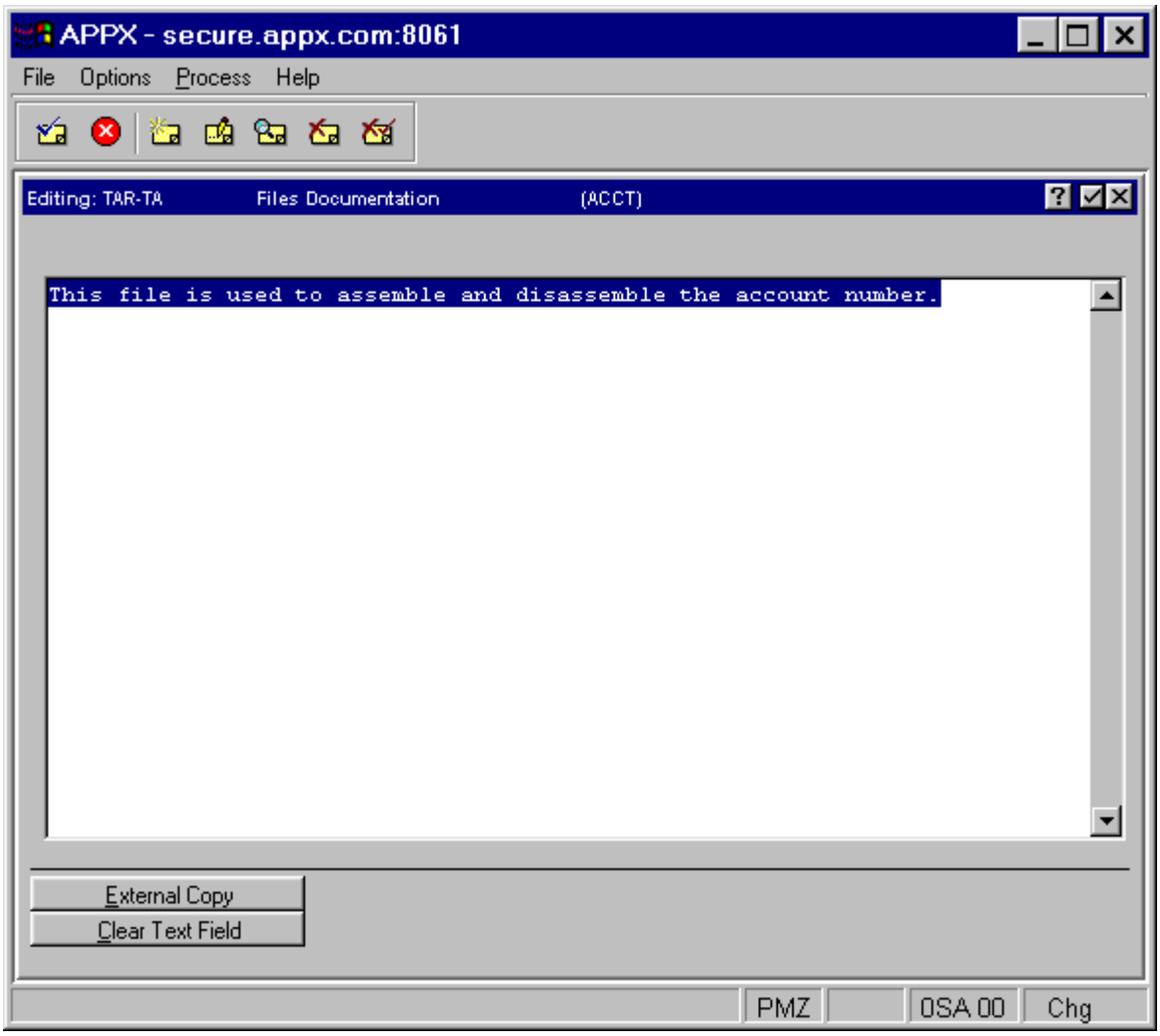


Figure 4-4. Edit Item Documentation Screen (for Files)

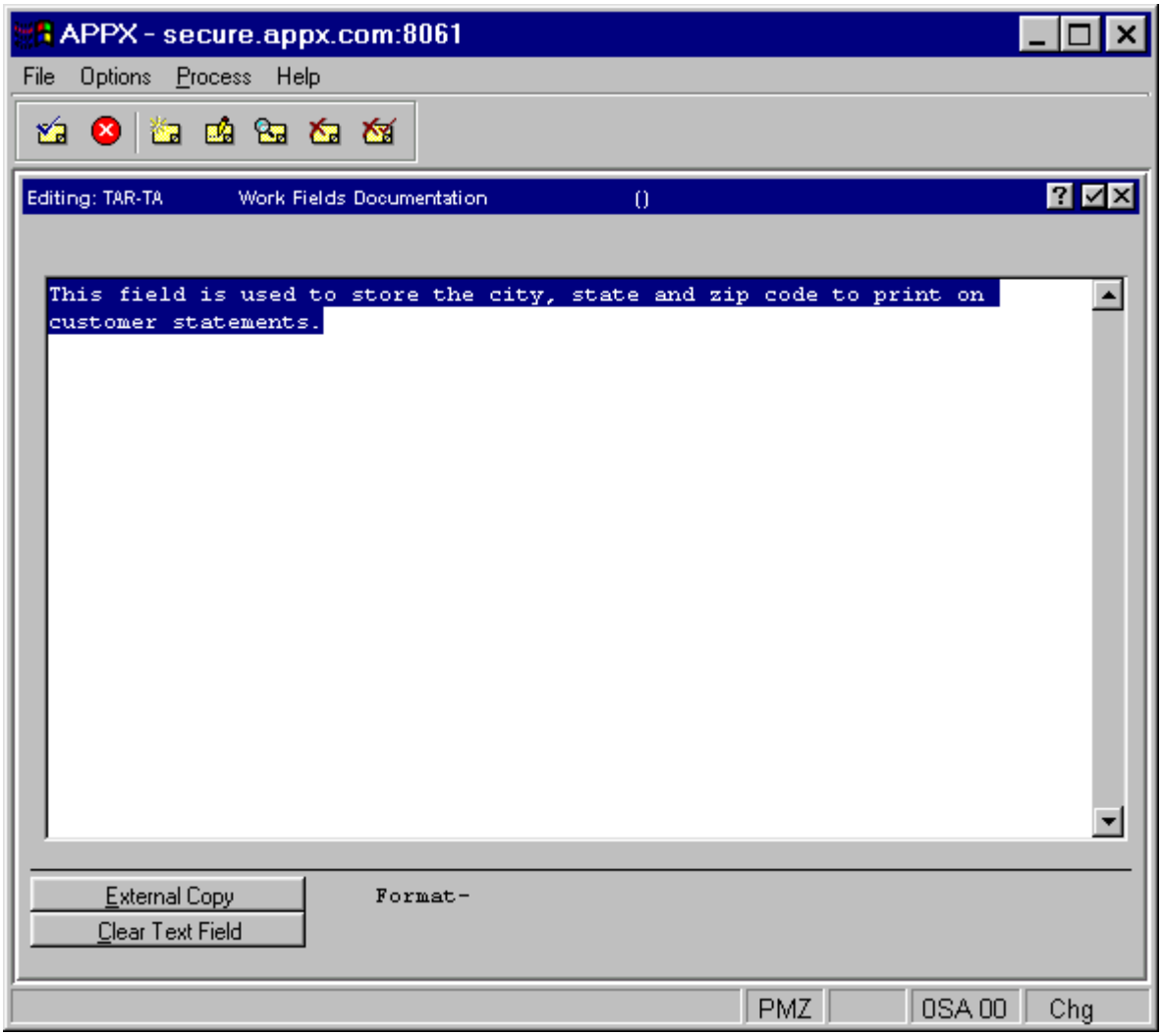


Figure 4-5. Edit Item Documentation Screen (for Work Fields)

The following two options are available on the Edit Item Documentation screen:

External Copy

Allows you to copy documentation from another item.

Clear Text Field

Clears all documentation currently entered for the item and places the cursor at the beginning of the field.

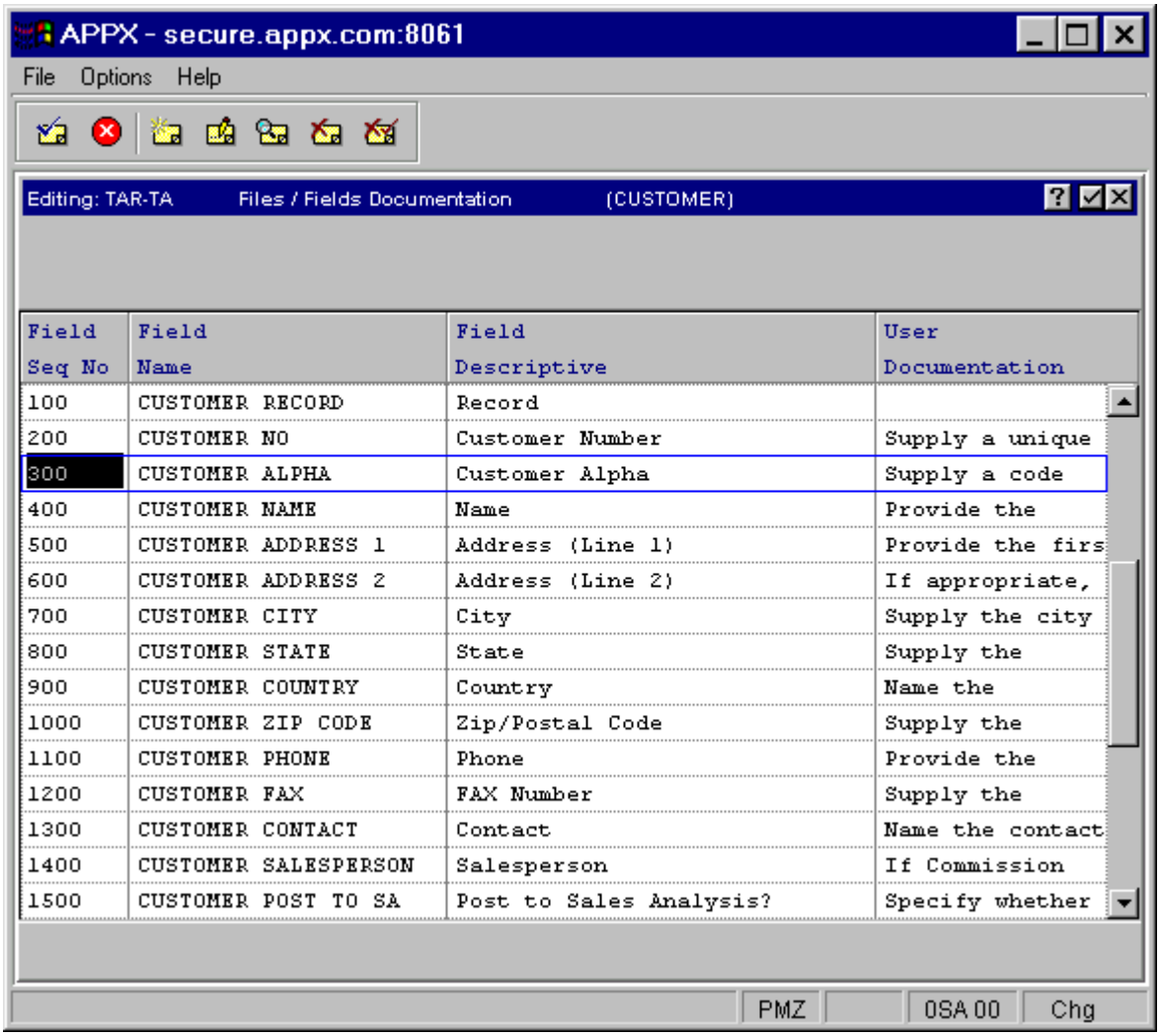
If the item is a domain or work field, format information is also displayed at the bottom of the screen. When you press ENTER for these types of items, you return to the Select Item Documentation screen.

When you press ENTER for a file, the fields for that file are displayed as discussed in the next section.

Documenting Fields

When you press ENTER from the Edit Item Documentation screen for a file, the fields for that file are displayed so you can edit their documentation as well. *Figure 4-6* shows the Select Field Documentation screen, and *Figure 4-7* shows the Edit Field Documentation screen:

Select Field Documentation screen



Field Seq No	Field Name	Field Descriptive	User Documentation
100	CUSTOMER RECORD	Record	
200	CUSTOMER NO	Customer Number	Supply a unique
300	CUSTOMER ALPHA	Customer Alpha	Supply a code
400	CUSTOMER NAME	Name	Provide the
500	CUSTOMER ADDRESS 1	Address (Line 1)	Provide the firs
600	CUSTOMER ADDRESS 2	Address (Line 2)	If appropriate,
700	CUSTOMER CITY	City	Supply the city
800	CUSTOMER STATE	State	Supply the
900	CUSTOMER COUNTRY	Country	Name the
1000	CUSTOMER ZIP CODE	Zip/Postal Code	Supply the
1100	CUSTOMER PHONE	Phone	Provide the
1200	CUSTOMER FAX	FAX Number	Supply the
1300	CUSTOMER CONTACT	Contact	Name the contact
1400	CUSTOMER SALESPERSON	Salesperson	If Commission
1500	CUSTOMER POST TO SA	Post to Sales Analysis?	Specify whether

Figure 4-6

Edit Field Documentation screen

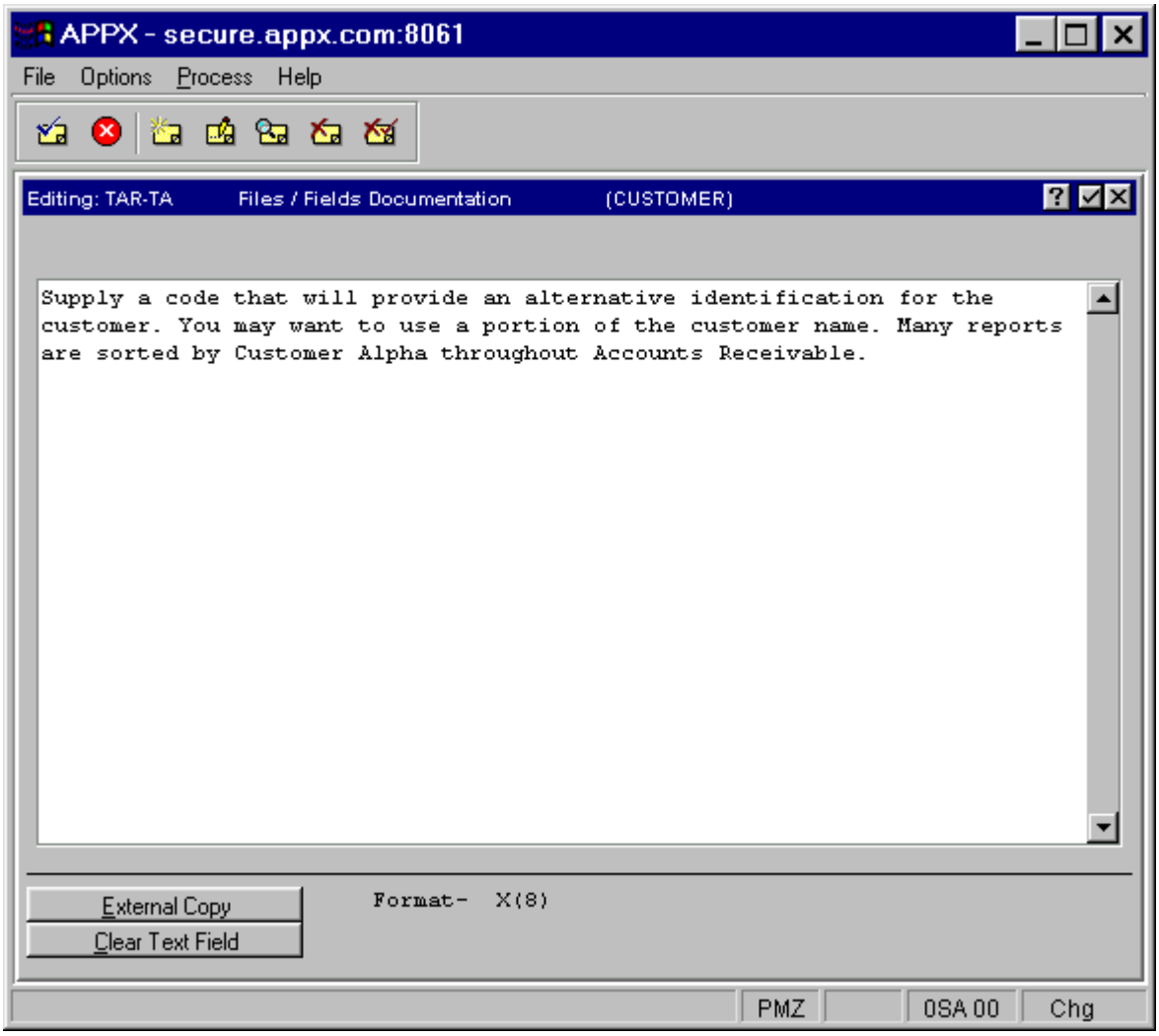


Figure 4-7

The following two options are available on the Edit Field Documentation screen:

External Copy

Allows you to copy documentation from another item.

Clear Text Field

Clears all documentation currently entered for the item and places the cursor at the beginning of the field.

Format information for the field is also available at the bottom of the screen.

Process Documentation Management

When you select the PROCESS MANAGEMENT option from the 'Manage Application Documentation' menu, the Process Documentation Management overlay appears as shown in *Figure 4-8*.

Process Documentation Management screen

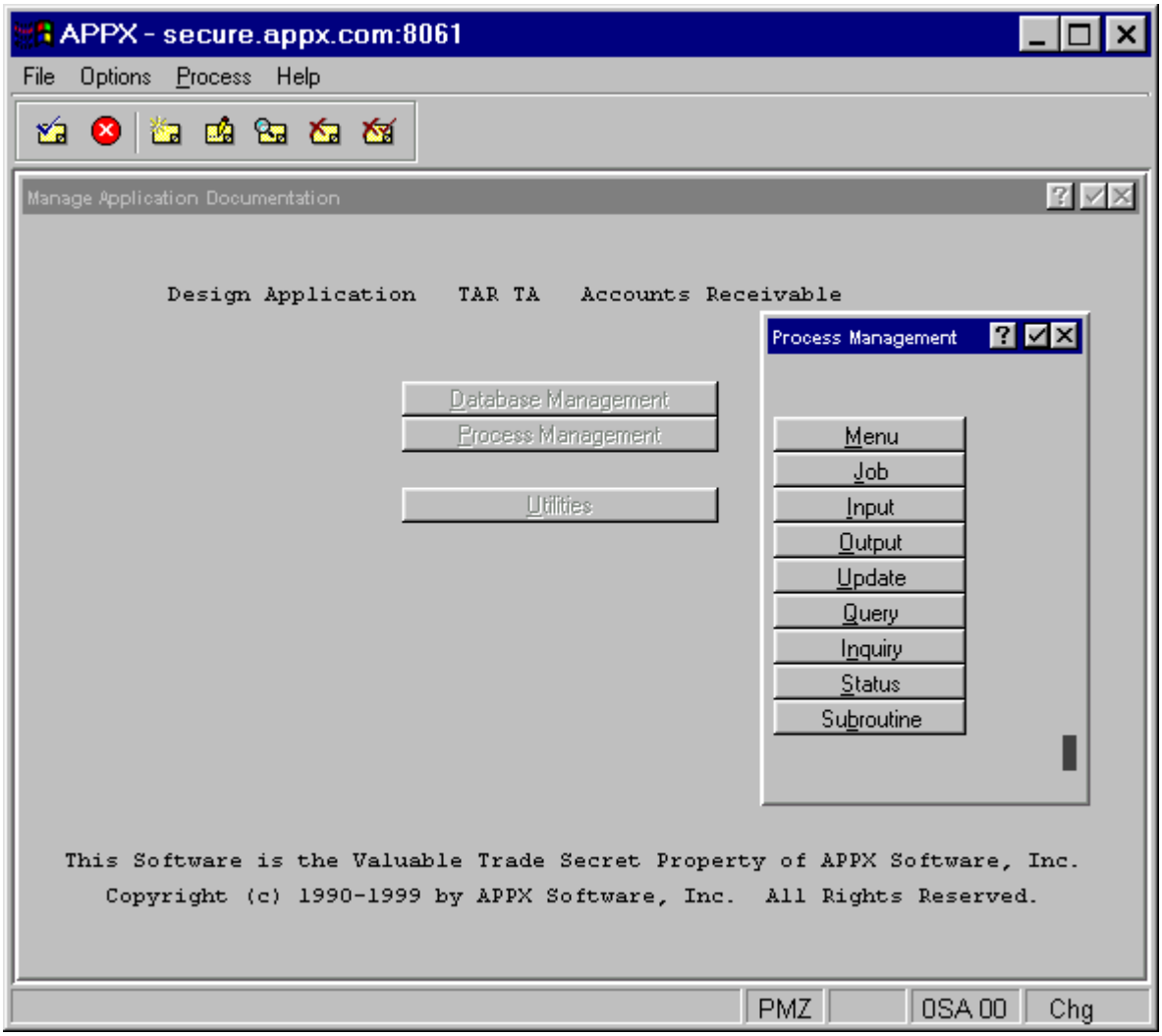


Figure 4-8

The options on the Process Documentation Management overlay provide access to documentation for the different types of processes. Choose one of these options to see the Select Process Documentation screen discussed in the next section.

Select Process Documentation Screen

When you choose the option for the type of process you wish to document from the Process Documentation Management overlay, a Select Process Documentation screen appears similar to the one shown in *Figure 4-9* below for input processes.

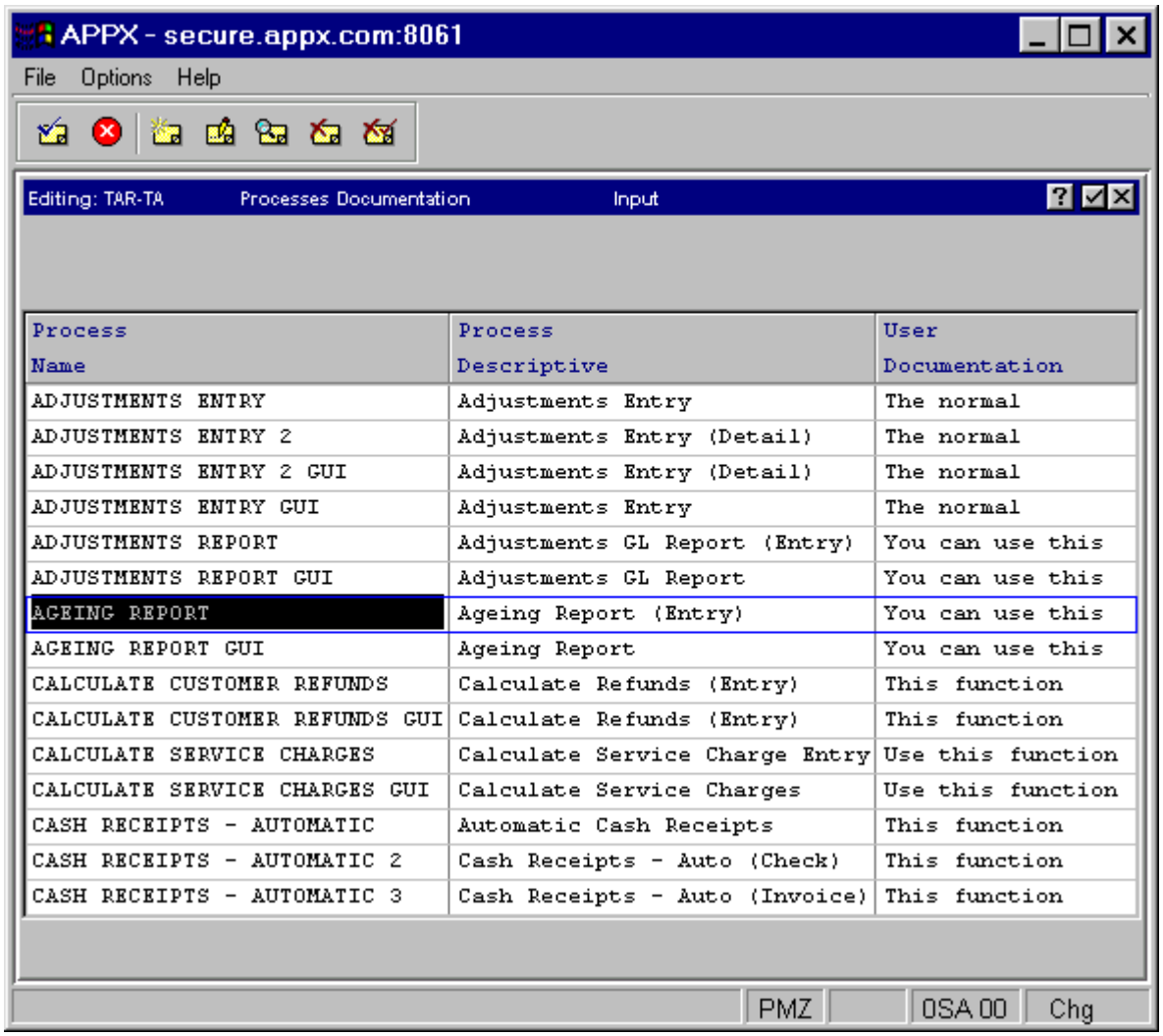


Figure 4-9. Select Process Documentation Screen (for Input Processes)

The Select Process Documentation screen presents a scrolling display of processes of the type you previously chose. Place your cursor on the specific process you wish to document and press ENTER to access the Edit Process Documentation screen discussed in the next section.

Edit Process Documentation Screen

When you choose the process you wish to document from the Select Process Documentation screen, an Edit Process Documentation screen appears similar to the one shown in *Figure 4-10* for input processes. This screen will display any documentation that has already been entered for the process or, if no documentation has been entered, the screen will be blank.

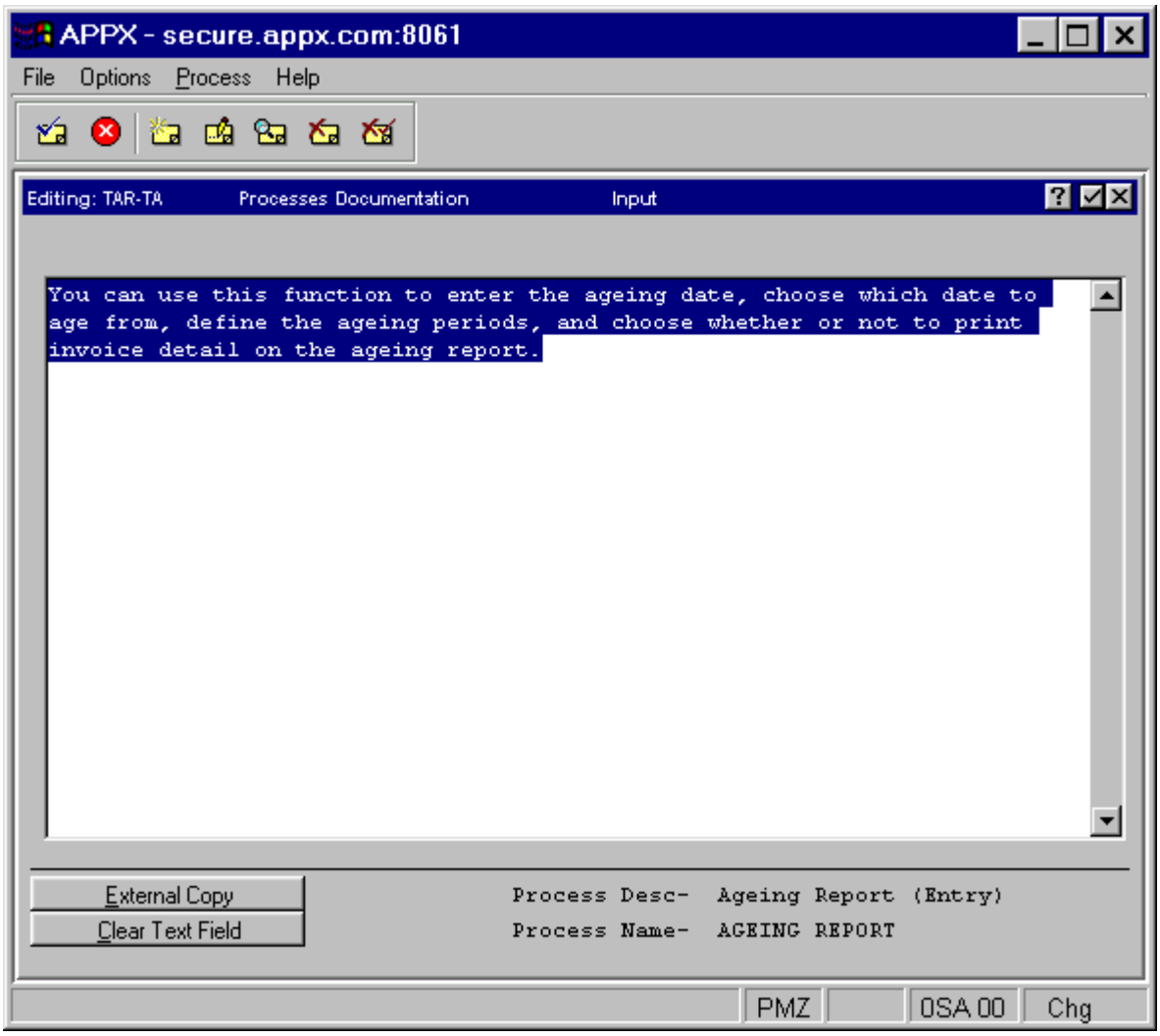


Figure 4-10. Edit Process Documentation Screen (for Input Processes)

The following two options are available on the Process Documentation Entry screen:

External Copy

Allows you to copy documentation from another process.

Clear Text Field

Clears all documentation currently entered for the process and places the cursor at the beginning of the field.

External Copy of Documentation

When you select the EXTERNAL COPY option from the Edit Item Documentation screen, the Edit Field Documentation screen, or the Edit Process Documentation screen, the External Copy of Documentation screen appears as shown in *Figure 4-11* below.

External Copy of Documentation screen

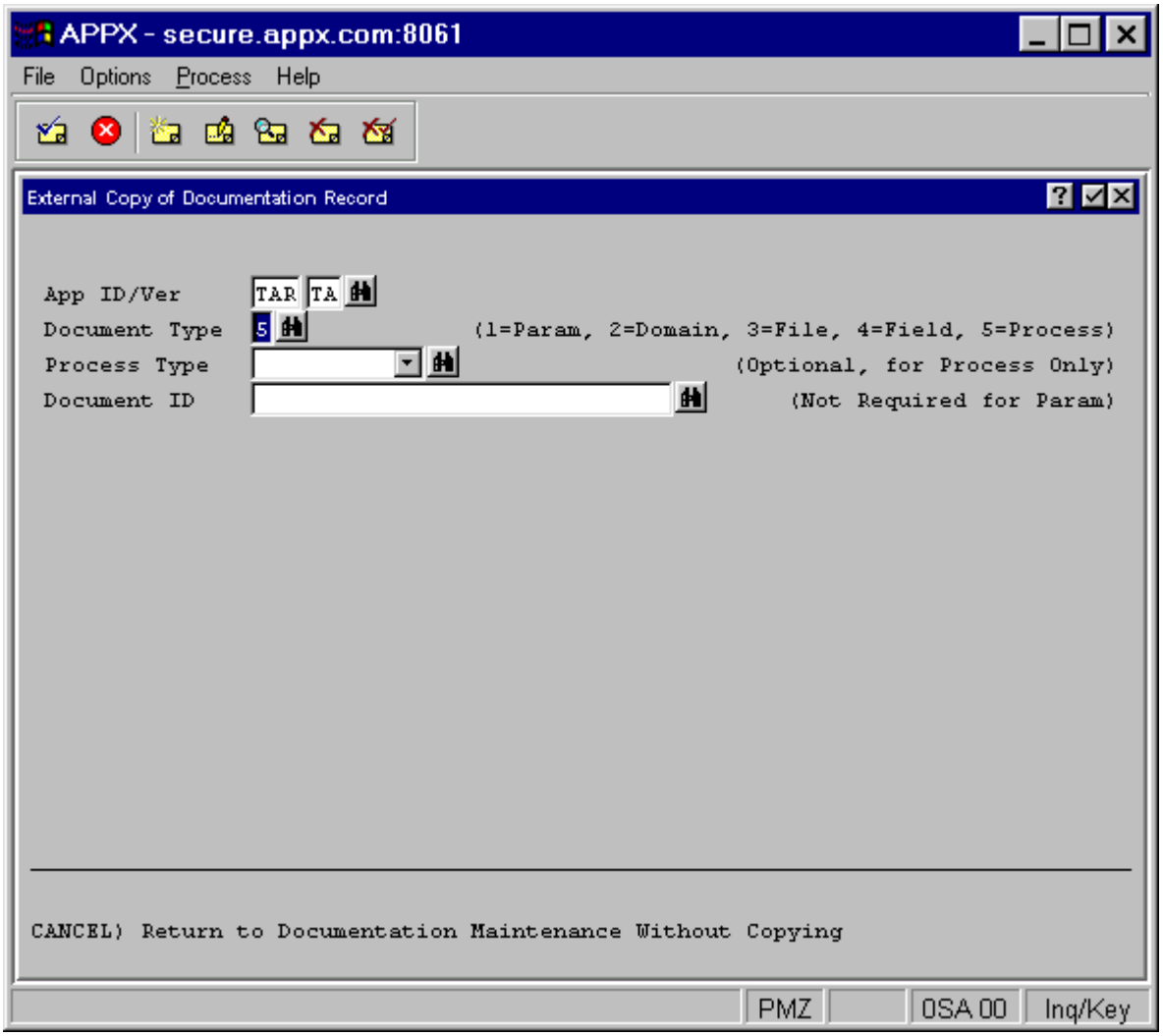


Figure 4-11

The External Copy of Documentation screen contains the following fields:

App ID/Ver

Provides the application/version where the desired documentation is located.

Document Type

Indicates the type of documentation to be copied
(1=Param, 2=Domain, 3=File, 4=Field or Work Field, 5=Process)

Process Type

Specifies the type of process containing the desired documentation and applies only to *Document Type* '5' (Process).

Document ID

Identifies the desired documentation entry and can be found by using SCAN.
Leave this field blank for *Document Type* '1' (Param).

After entering the appropriate information, press ENTER and the documentation entry for the item or process you specified will appear on the screen. Press ENTER again and the text will appear on the Edit Documentation screen from which you originally invoked EXTERNAL COPY.

Press CANCEL to abort the External Copy and return to the original documentation screen without copying.

Documentation Management Utilities

When you select the UTILITIES option from the 'Manage Application Documentation' menu, the overlay shown in *Figure 4-12* displays.

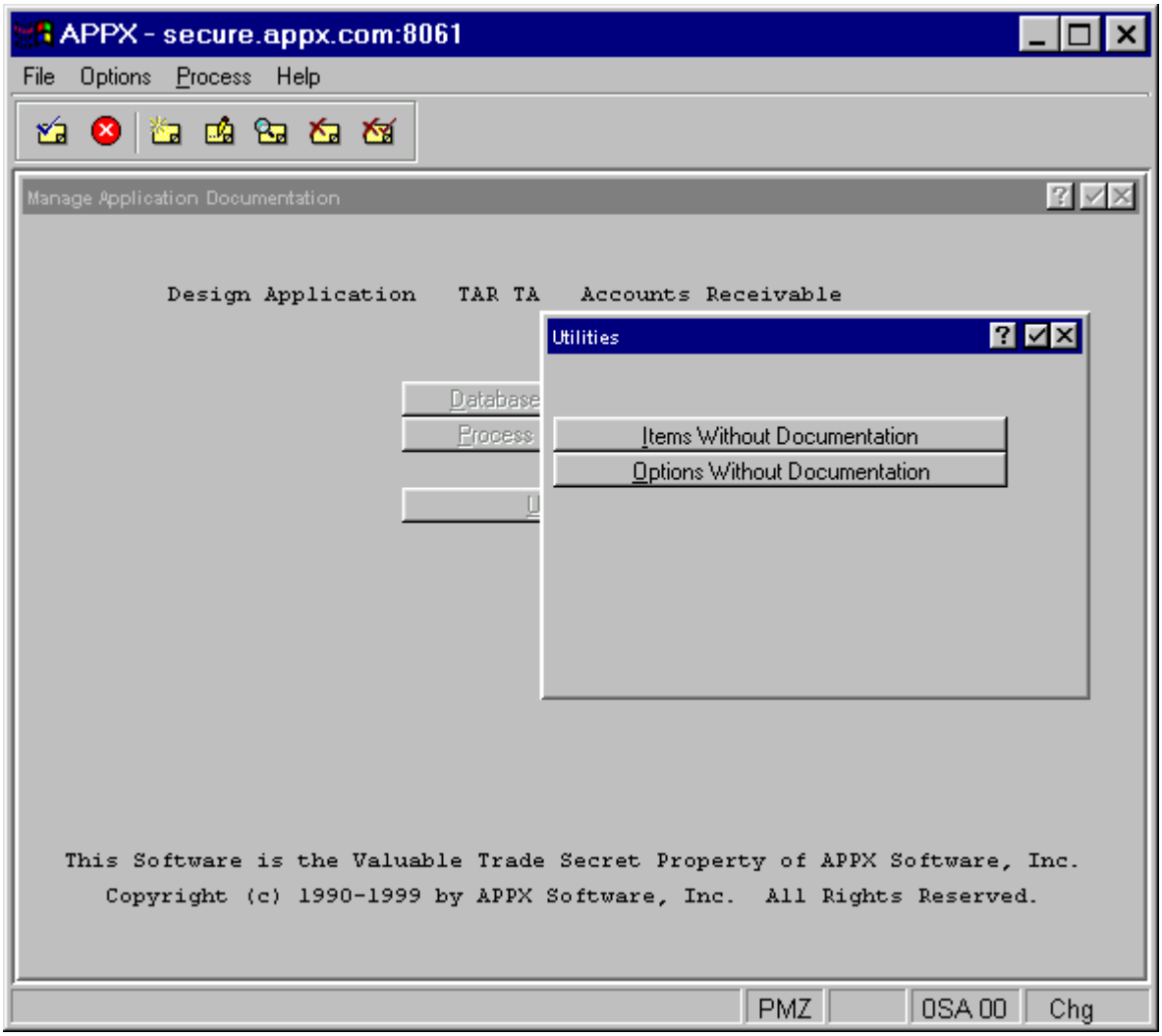


Figure 4-12. Documentation Management Utilities Overlay

The utilities available on this overlay are listed below:

- *Items Without Documentation*
- *Options Without Documentation*

These utilities are described in the following sections of this manual.

Items Without Documentation

When you select the ITEMS WITHOUT DOCUMENTATION option from the Documentation Management Utilities overlay, you will be generating a list, like the one shown in *Figure 4-13*, of all items on interactive screens that have not yet been documented.

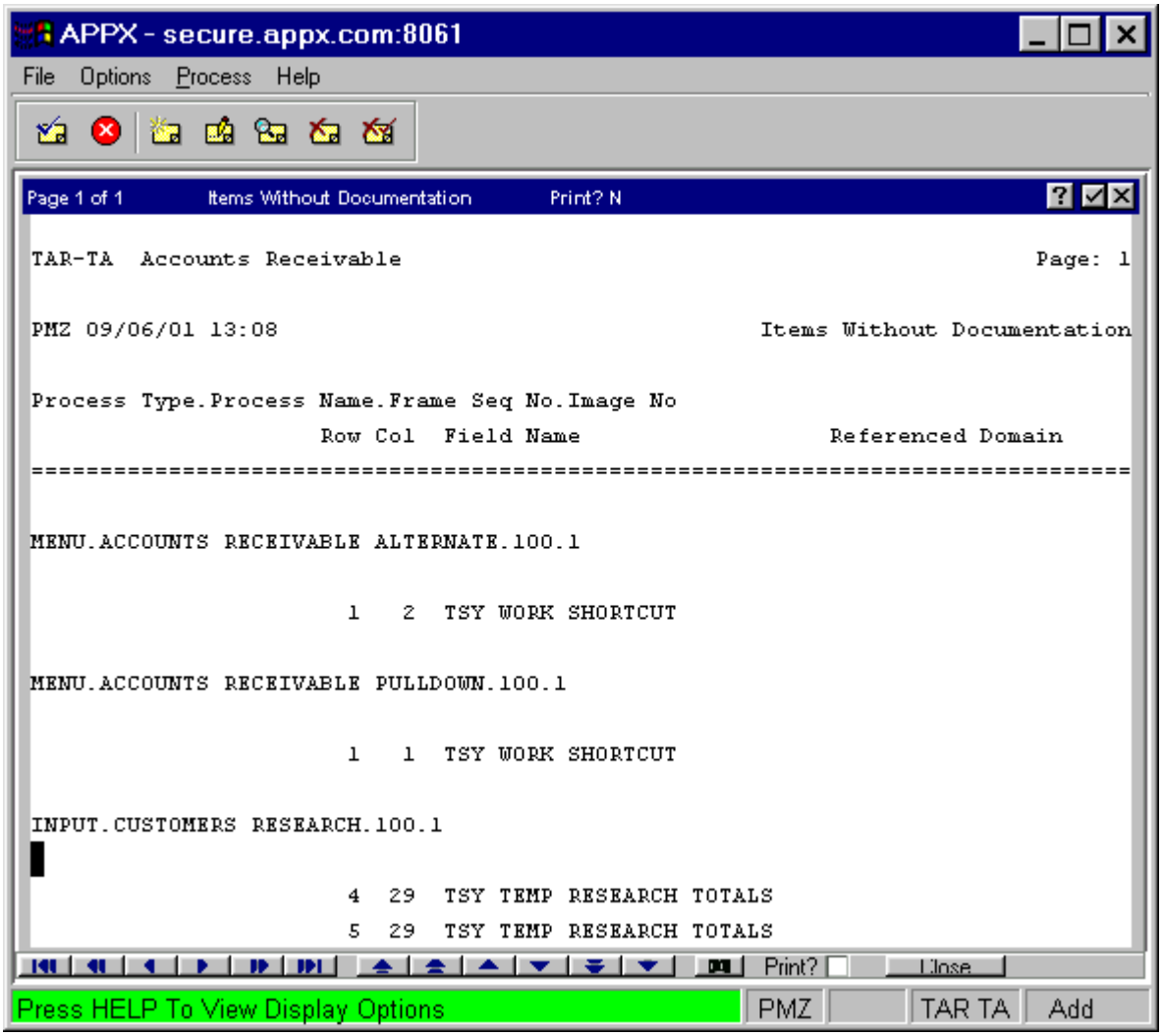


Figure 4-13. Items Without Documentation List

Options Without Documentation

When you select OPTIONS WITHOUT DOCUMENTATION from the Documentation Management Utilities overlay, you will be generating a list, like the one shown in *Figure 4-14*, of all options on interactive screens that have not yet been documented.

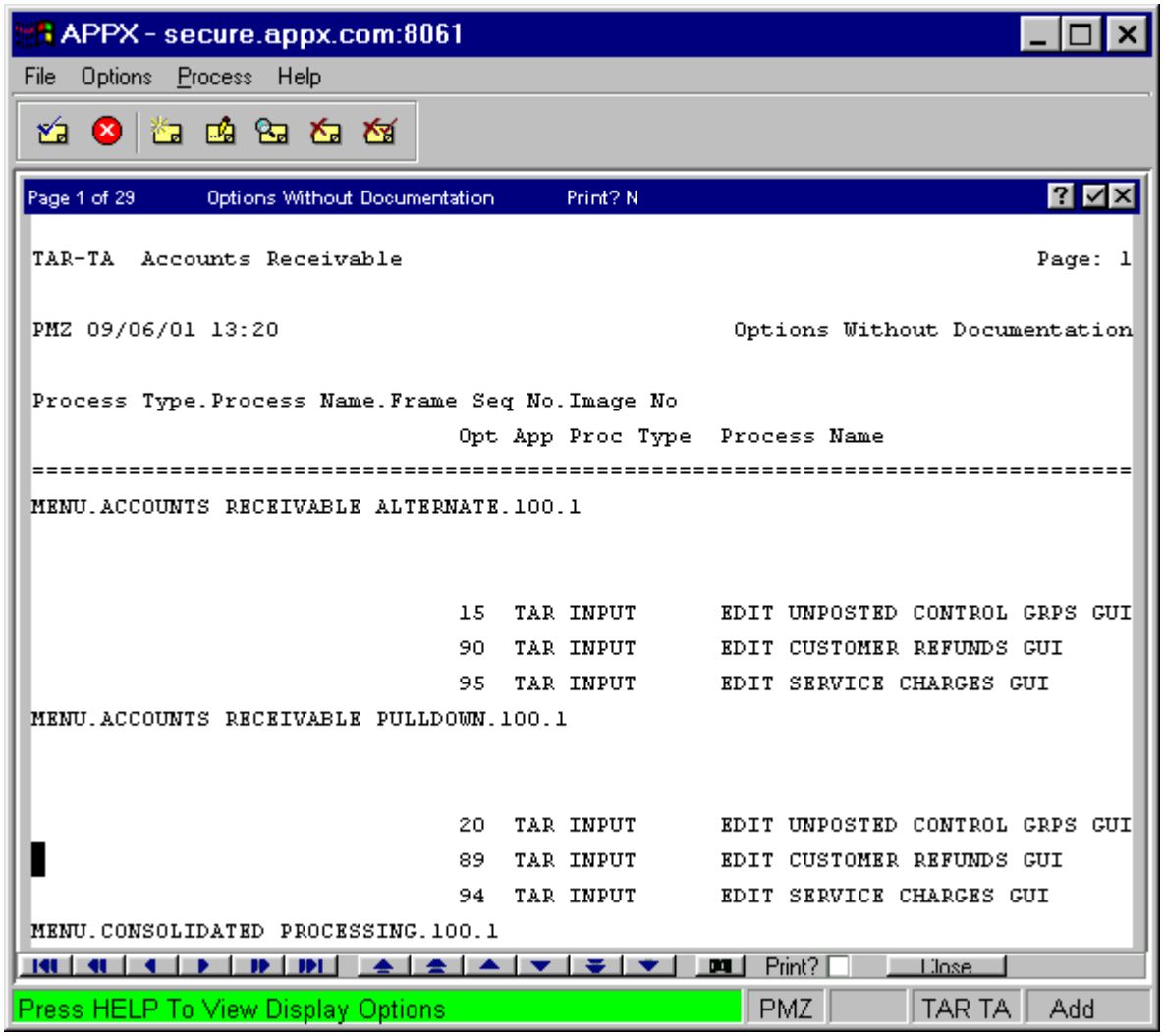


Figure 4-14. Options Without Documentation List

CHAPTER 5

Application Security Management

'Manage Application Security' Menu

The purpose of application security management is to assign security codes as desired to files, fields, and processes in the selected application. User rights to these security codes are granted according to the user's security profile defined in Users File Maintenance in System Administration. Also see the sections on APPX Security and Security Profiles File Maintenance in Chapter 2 of this manual for more information.

When you select SECURITY MANAGEMENT from the APPX Main Menu (or MANAGE APPLICATION SECURITY from the Security Menu in System Administration), the 'Manage Application Security' Menu shown in *Figure 5-1* appears:

'Manage Application Security' menu

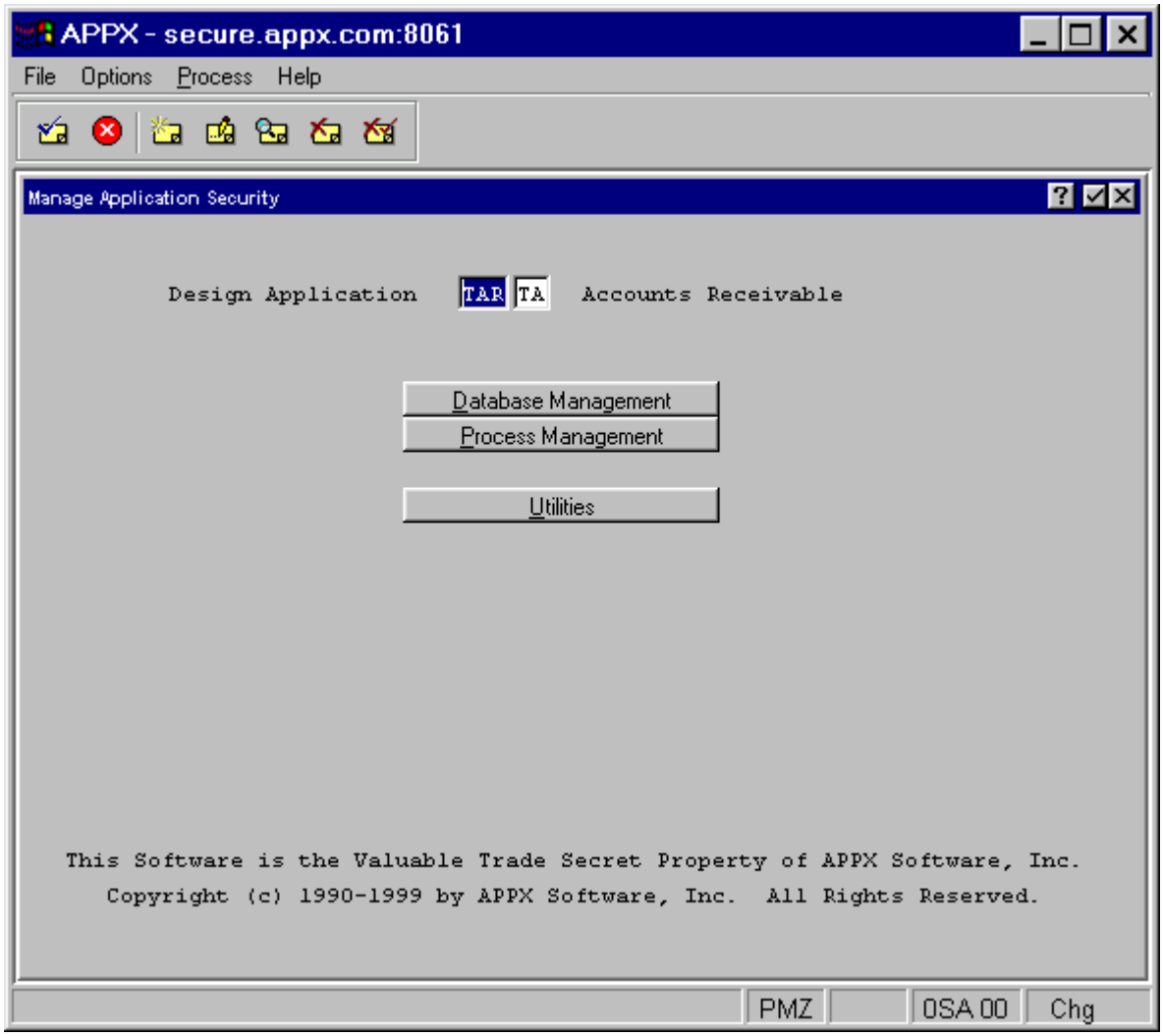


Figure 5-1

Enter the application ID and version number of the application for which you want to edit security, and press ENTER. Then choose one of the following options:

- *Database Management* displays a menu from which you can edit security codes for the domains, files and related fields, and work fields in the selected application.
- *Process Management* displays a menu from which you can edit security codes for all types of processes in the selected application.
- *Utilities* displays a menu containing options for you to view, clear, archive and post all security codes within the selected application.

The sections that follow describe each of these options in detail. Use these options to connect the security codes set up in Security Profile maintenance to individual fields and processes in each of the applications on your system

Reminder: All APPX Security Codes are 4-digit numbers from 0000 to 9999. Leading zeros must be entered.

Database Security Management

When you select DATABASE MANAGEMENT from the 'Manage Application Security' menu, the Database Security Management overlay shown in *Figure 5-2* appears:

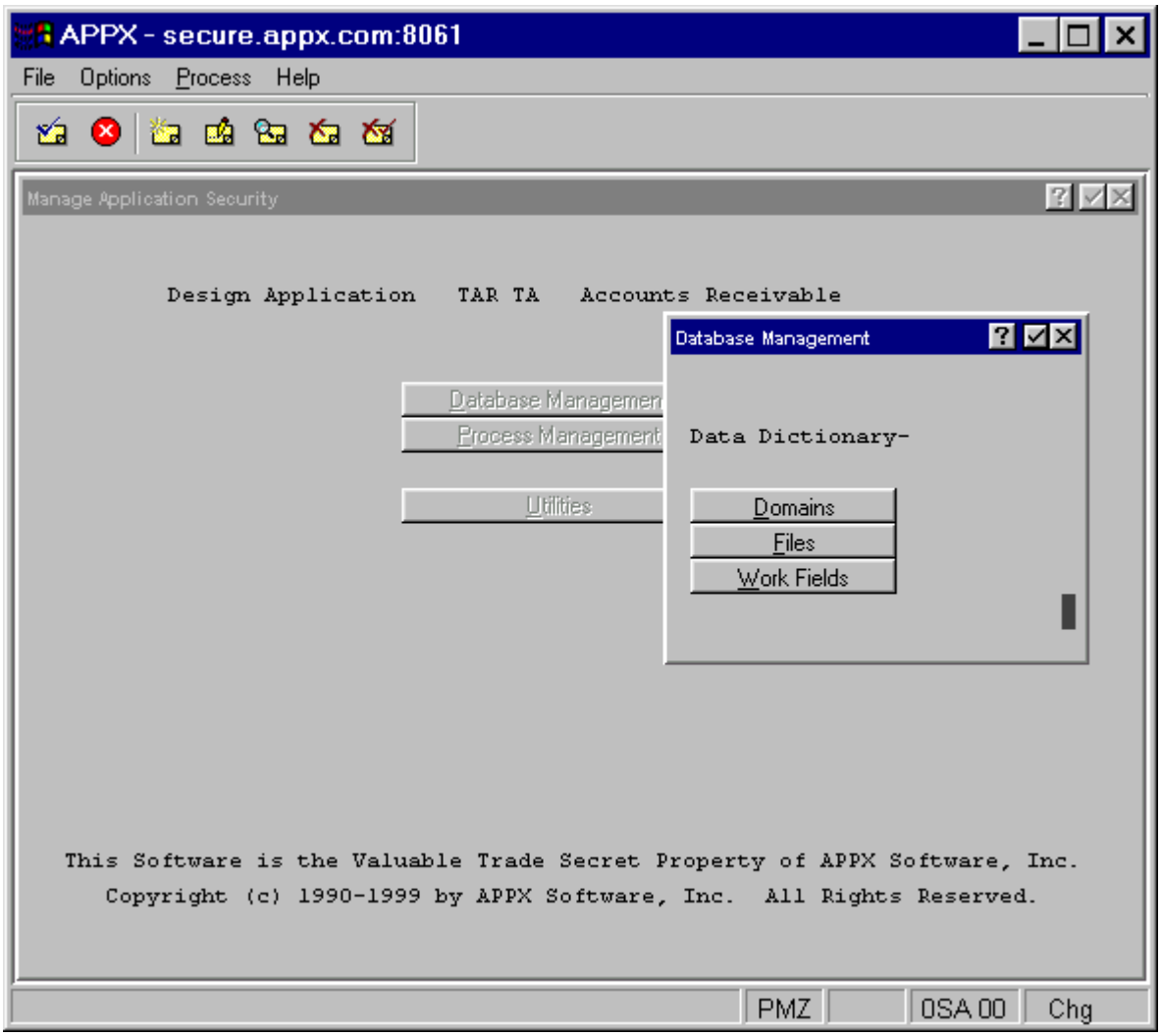


Figure 5-2. Database Security Management Overlay

The options on the Database Security Management overlay provide access to security for the different types of database items as follows:

- *Domains*
- *Files* (and Fields)
- *Work Fields*

These options are described in detail later in this manual. In general, however, when you select an option, a list of all items of that particular type in the selected application displays. From this list you can define “read” and “modify” rights for every item on the list by placing your cursor on the row containing the item name. In general, a ***Read*** right lets a user see the contents of a field, but not change it; a ***Modify*** right lets a user change the contents of a field. *Modify* rights always include *Read* rights.

There is a hierarchy in assigning security rights to fields. If a field is a domain type field and its domain is assigned a security code, no security code can be assigned to the field itself. If no security code is assigned to the domain, a security code can be assigned to the field. If a field is not secured by domain or field security, it is subject to the security code assigned at the file level which provides security to all unsecured fields in that file. If no security codes are assigned at any of these levels, access to the field is unrestricted.

Domains Security

When you select the DOMAINS option from the Database Security Management overlay, a screen appears with a list of all domains in the selected application as shown in *Figure 5-3*.

Domains Security Definition screen

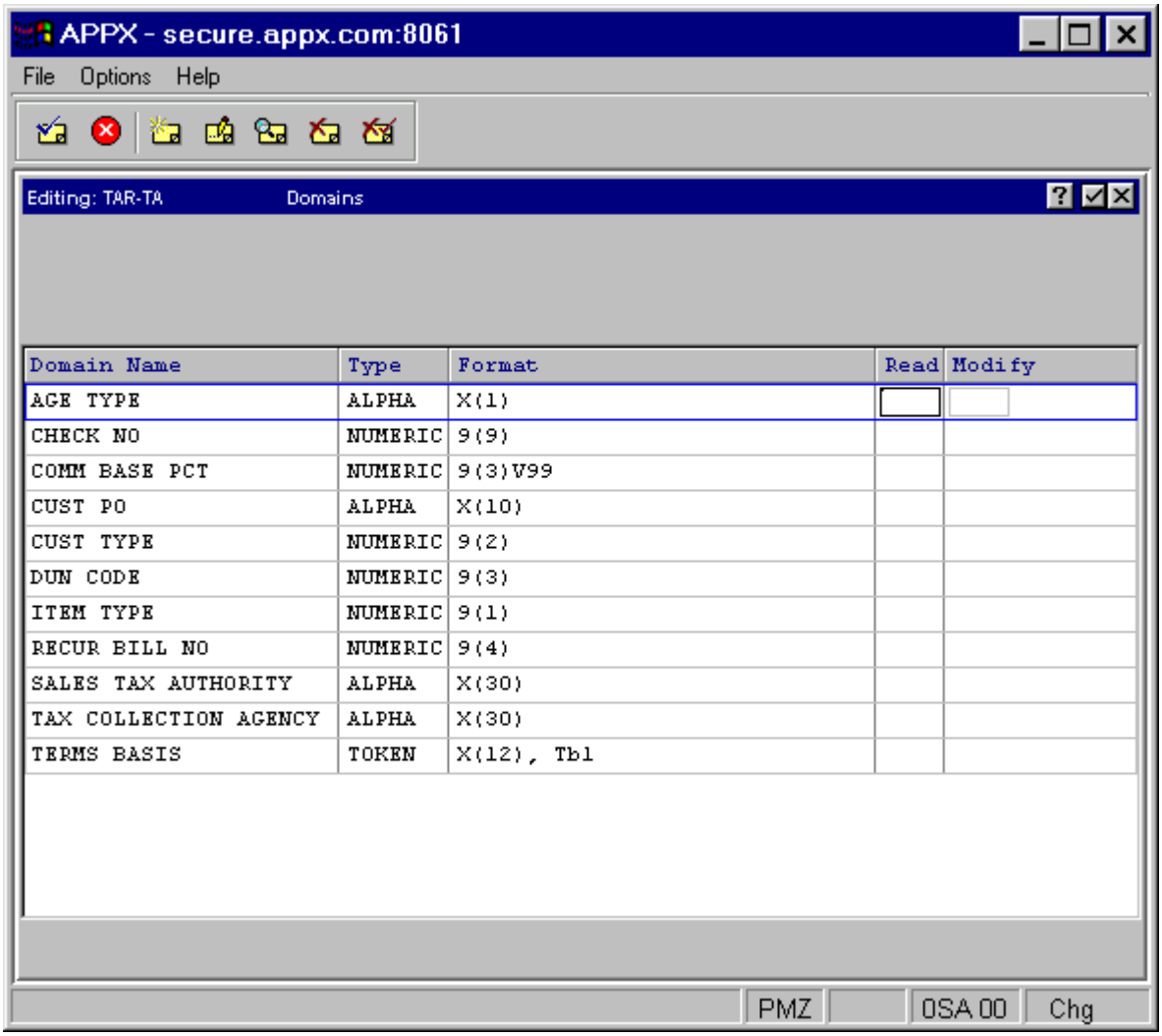


Figure 5-3

The following fields are available for input:

Read

Specifies the security code that must be included in a user's profile for the user to see the contents of any field that references the domain. ***This code cannot be overridden at the field level.***

Modify

Specifies the security code that must be included in a user's profile for the user to modify any field that references the domain. ***This code cannot be overridden at the field level.***

Files Security

When you select the FILES option from the Database Security Management overlay, a screen appears with a list of all files in the selected application as shown in *Figure 5-4*.

Files Security Definition screen

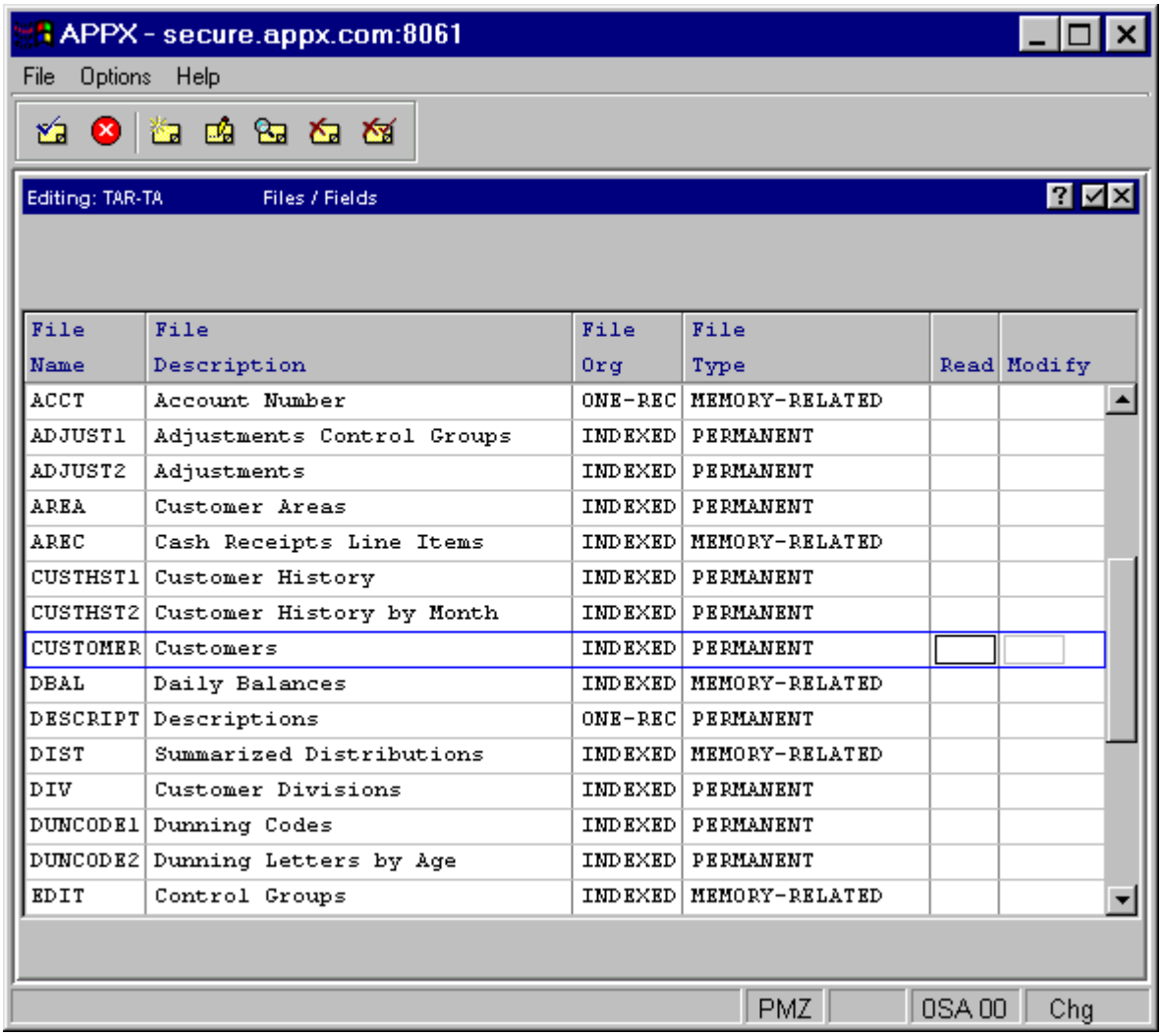


Figure 5-4

The following fields are available for input:

Read

Specifies the security code that must be included in a user's profile for the user to see the contents of unsecured fields in this file.

Modify

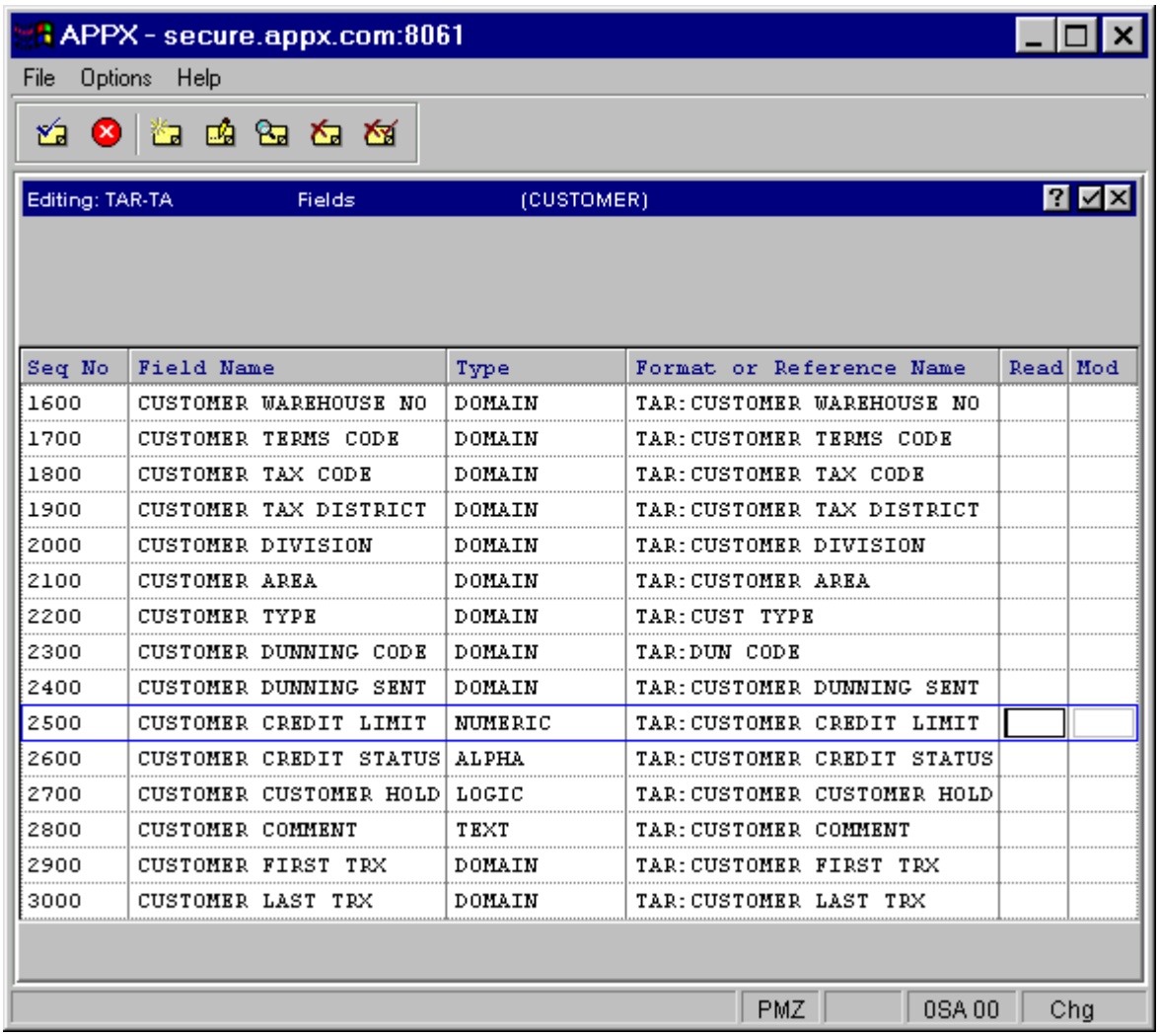
Specifies the security code that must be included in a user's profile for the user to modify the contents of unsecured fields in this file.

If these fields are blank, fields in this file are not secured unless domain or field level security is present. To enter security at the field level, select the file and press ENTER.

Fields Security

When you press ENTER from the Files Security definition screen, a screen appears which displays all fields within the selected file as shown in *Figure 5-5*. Each field can be edited for security unless it is protected by a domain security code.

Fields Security Definition screen



Seq No	Field Name	Type	Format or Reference Name	Read	Mod
1600	CUSTOMER WAREHOUSE NO	DOMAIN	TAR:CUSTOMER WAREHOUSE NO		
1700	CUSTOMER TERMS CODE	DOMAIN	TAR:CUSTOMER TERMS CODE		
1800	CUSTOMER TAX CODE	DOMAIN	TAR:CUSTOMER TAX CODE		
1900	CUSTOMER TAX DISTRICT	DOMAIN	TAR:CUSTOMER TAX DISTRICT		
2000	CUSTOMER DIVISION	DOMAIN	TAR:CUSTOMER DIVISION		
2100	CUSTOMER AREA	DOMAIN	TAR:CUSTOMER AREA		
2200	CUSTOMER TYPE	DOMAIN	TAR:CUST TYPE		
2300	CUSTOMER DUNNING CODE	DOMAIN	TAR:DUN CODE		
2400	CUSTOMER DUNNING SENT	DOMAIN	TAR:CUSTOMER DUNNING SENT		
2500	CUSTOMER CREDIT LIMIT	NUMERIC	TAR:CUSTOMER CREDIT LIMIT		
2600	CUSTOMER CREDIT STATUS	ALPHA	TAR:CUSTOMER CREDIT STATUS		
2700	CUSTOMER CUSTOMER HOLD	LOGIC	TAR:CUSTOMER CUSTOMER HOLD		
2800	CUSTOMER COMMENT	TEXT	TAR:CUSTOMER COMMENT		
2900	CUSTOMER FIRST TRX	DOMAIN	TAR:CUSTOMER FIRST TRX		
3000	CUSTOMER LAST TRX	DOMAIN	TAR:CUSTOMER LAST TRX		

Figure 5-5

The following fields are available for input:

Read

Specifies the security code that must be included in a user's profile for the user to see the contents of the field. If blank and not protected by a domain security code, APPX uses the *Read* security code entered for the file.

Modify

Specifies the security code that must be included in a user's profile for the user to modify the contents of the field. If blank and not protected by a domain security, APPX uses the *Modify* security code entered for the file.

Work Fields Security

When you select the WORK FIELDS option from the Database Security Management overlay, a screen appears with a list of all work fields in the selected application as shown in *Figure 5-6*.

Work Fields Security Definition screen

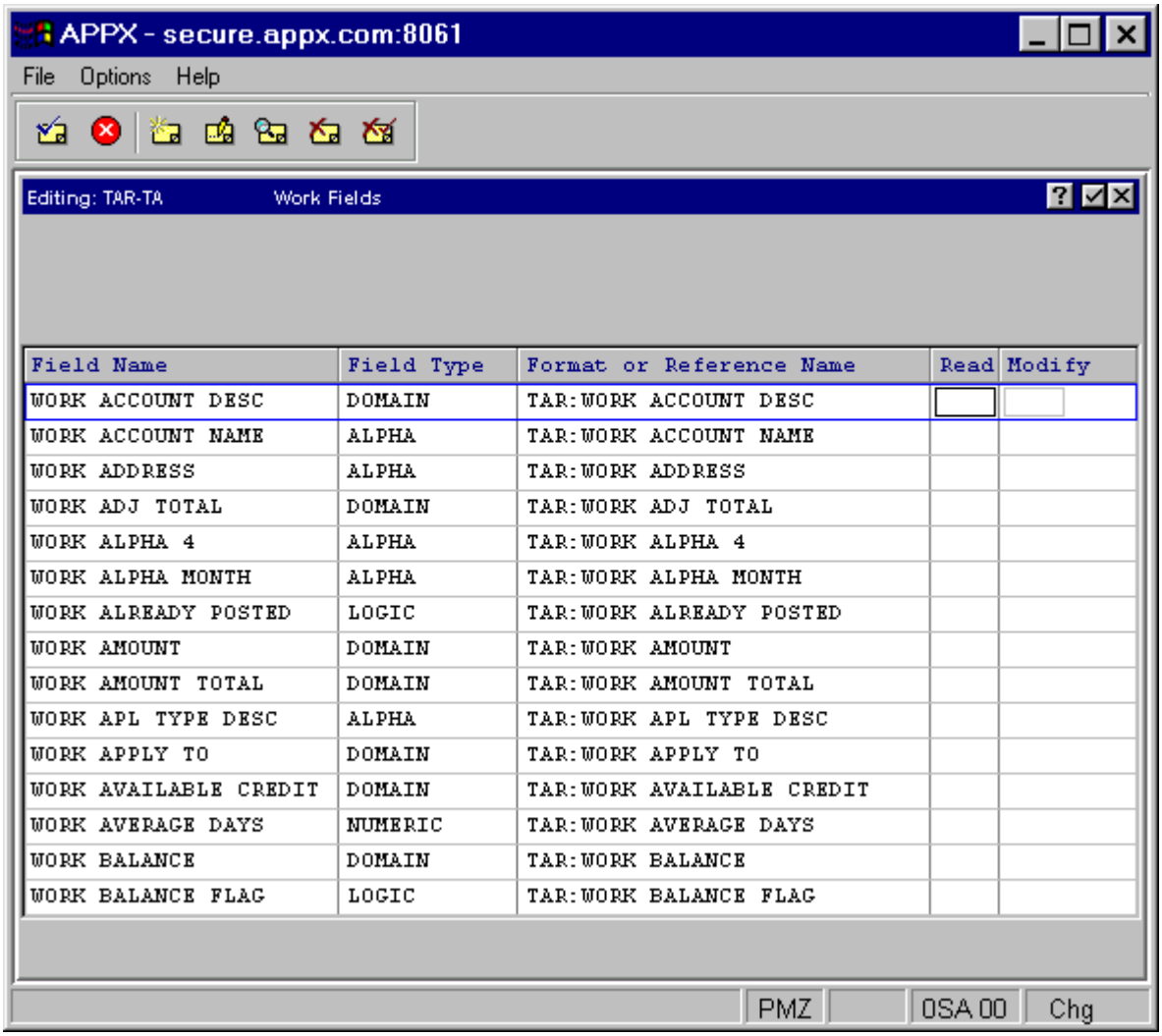


Figure 5-6

The following fields are available for input:

Read

Specifies the security code that must be included in a user's profile for the user to see the contents of the work field. If blank and unprotected by a domain security code, the field is unsecured.

Modify

Specifies the security code that must be included in a user's profile for the user to modify the contents of the work field. If blank and unprotected by a domain security code, the field is unsecured.

Process Security Management

When you select PROCESS MANAGEMENT from the 'Manage Application Security' menu, the Process Security Management overlay appears as shown in *Figure 5-7*.

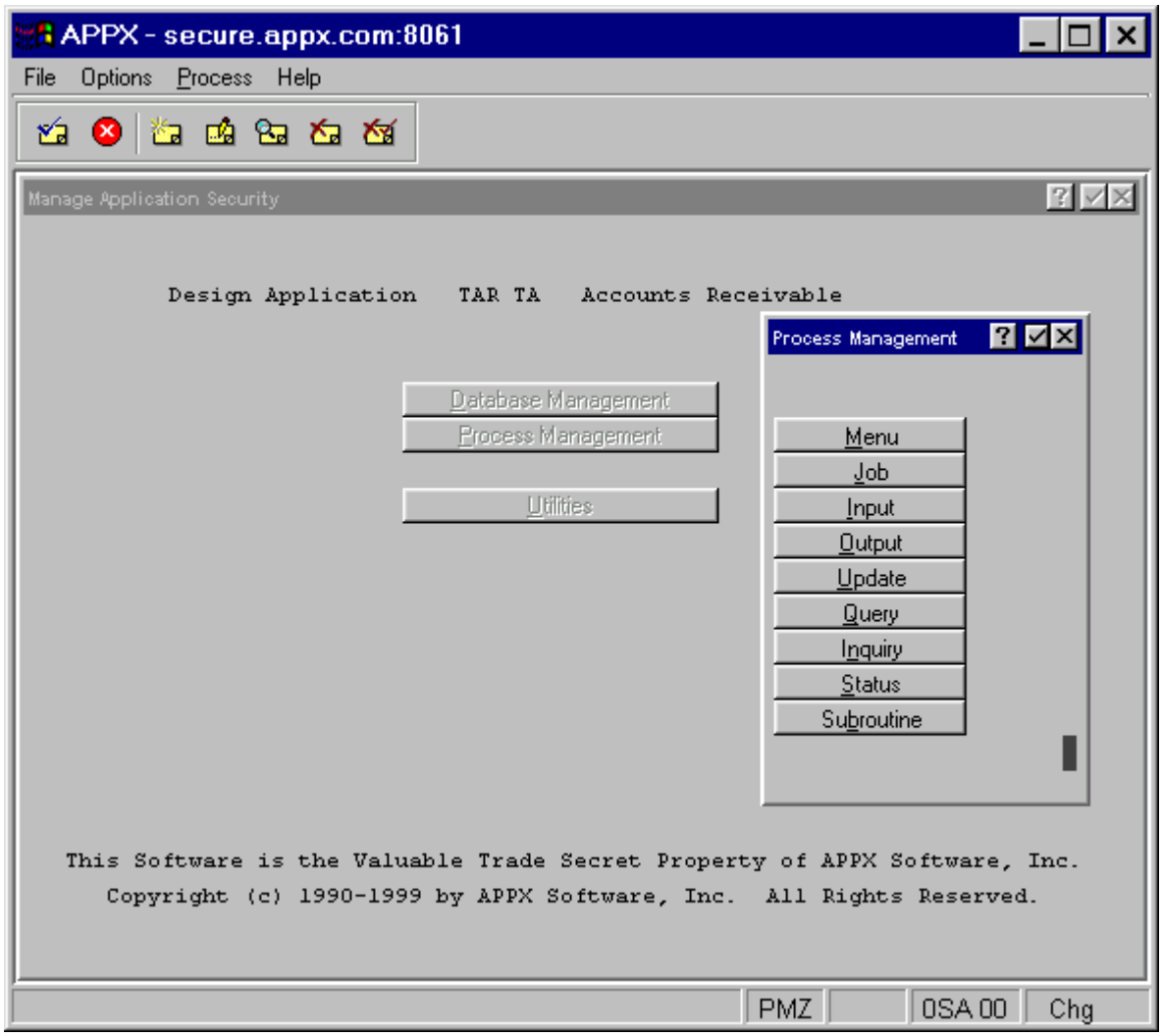


Figure 5-7. Process Security Management Overlay

Use the options on this overlay to choose the type of process for which you wish to edit security. Once you select an option, a screen appears that lists every process of that particular type in your application. For specific details, see Process Security for Input Processes and Process Security for Non-Input Processes as appropriate.

> **Note:** It is not possible to select the QUERY option or the SUBROUTINE option from the Process Security Management overlay because security management is not applicable to either.

Process Security for Non-Input

When you select an option other than INPUT from the Process Security Management overlay, a screen appears similar to the one shown in *Figure 5-8* for output processes:

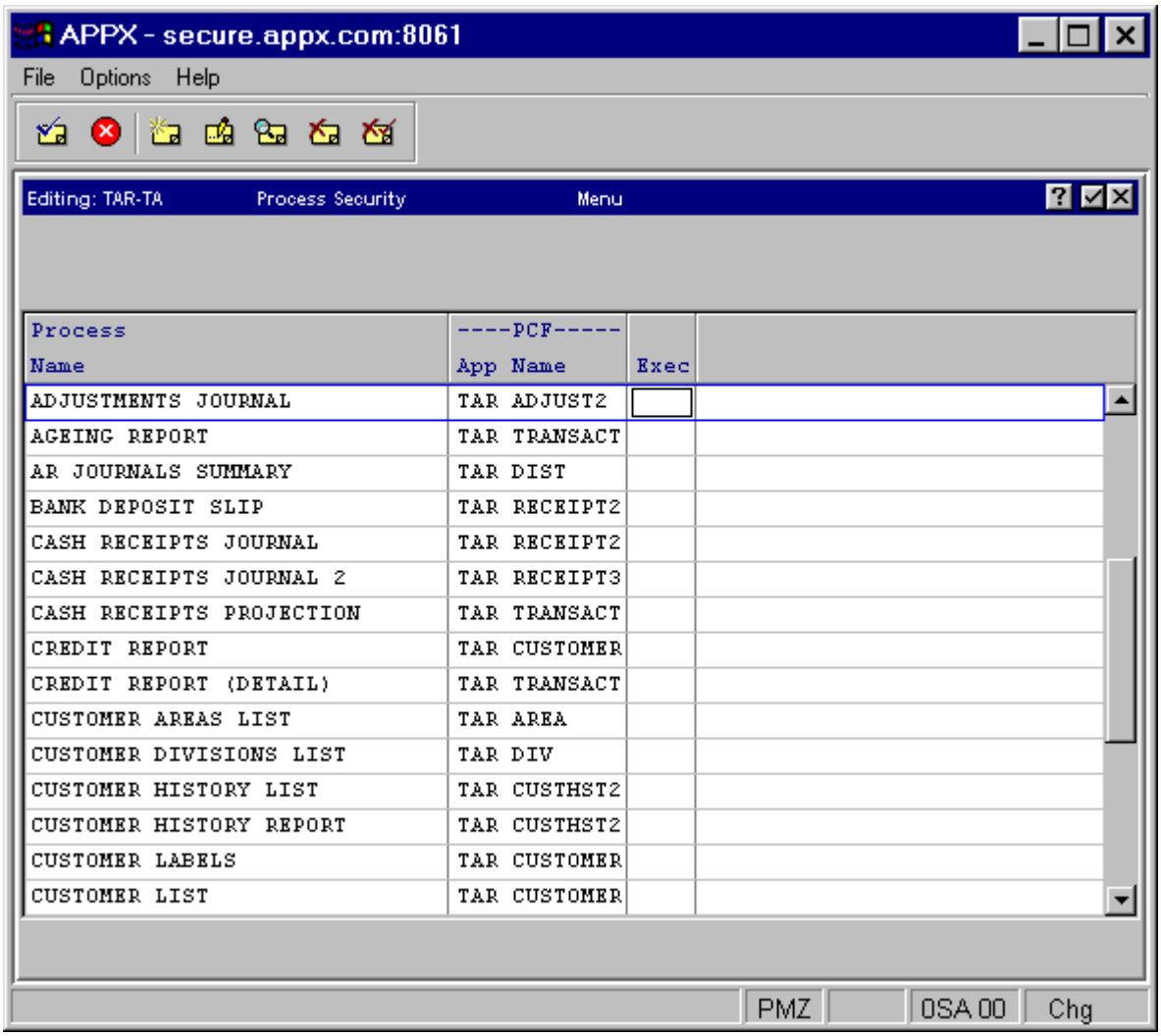


Figure 5-8. Process Security Definition Screen for Non-Input Processes

Once you have selected a process, the following field is available:

Exec

Specifies the security code that must be included in a user's profile for the user to run this process. If blank, there are no restrictions.

Process Security for Input

When you select the INPUT option from the Process Security Management overlay, a screen appears as shown in *Figure 5-9*.

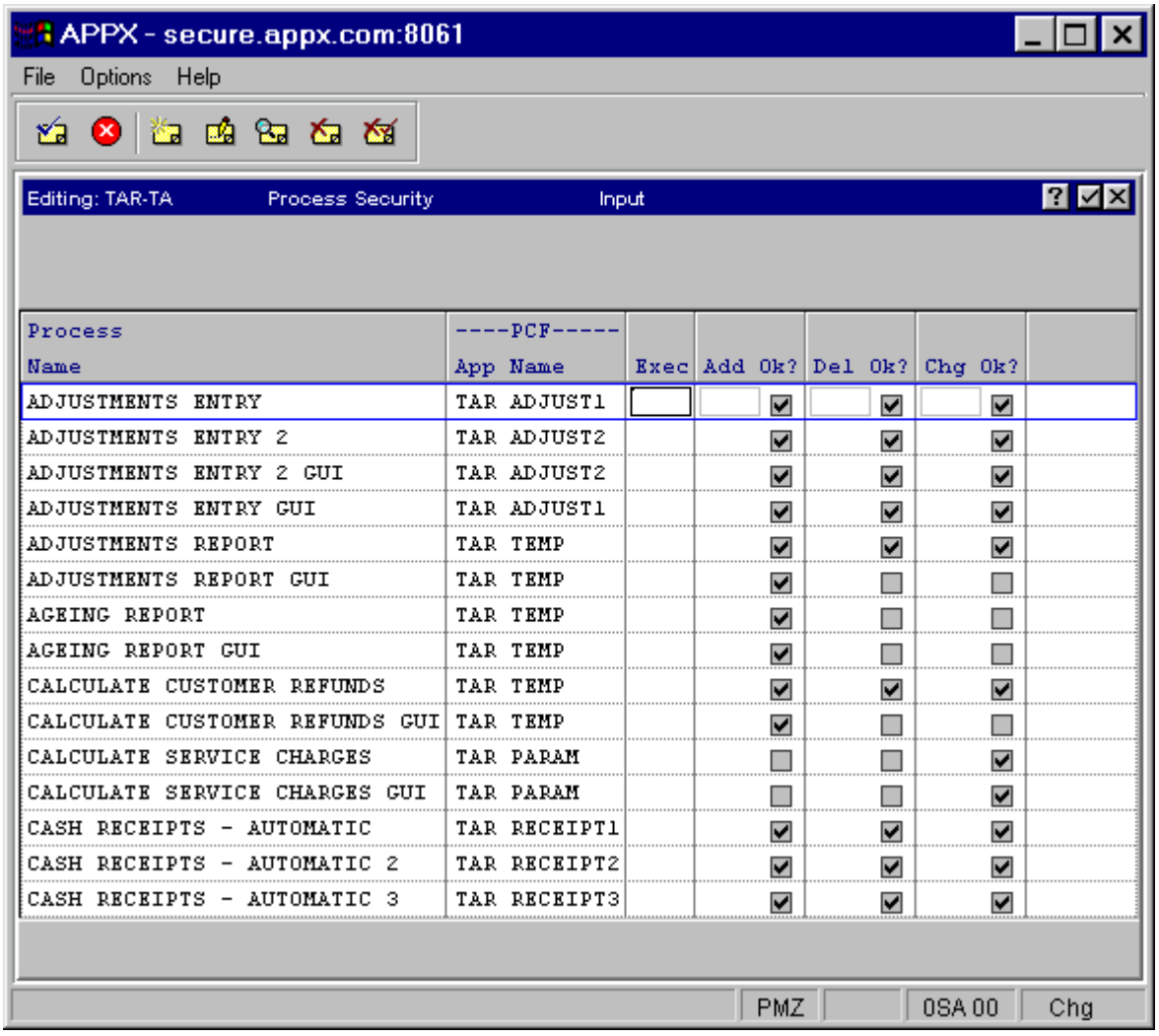


Figure 5-9. Process Security Definition Screen for Input Processes

Once you have selected a process, the following fields are available:

Exec

Specifies the security code that must be included in a user's profile for the user to run this process. If blank, there are no restrictions.

Add

Specifies the security code that must be included in a user's profile for the user to add records to a file with this process. If blank, there are no restrictions. *OK?* denotes the non-modifiable checkbox containing either "yes" ☑ or "no" ☐ to indicate whether or not the designer allowed any add rights for this process.

Del

Specifies the security code that must be included in a user's profile for the user to delete records from a file with this process. If blank, there are no restrictions. *OK?* denotes the non-modifiable checkbox containing either "yes" ☑ or "no" ☐ to indicate whether or not the designer allowed any delete rights for this process.

Chg

Specifies the security code that must be included in a user's profile for the user to change records in a file with this process. If blank, there are no restrictions. *OK?* denotes the non-modifiable checkbox containing either "yes" ☑ or "no" ☐ to indicate whether or not the designer allowed any change rights for this process.

Security Management Utilities

When you select the UTILITIES option from the 'Manage Application Security' menu, the Security Management Utilities overlay appears as shown in *Figure 5-10*:

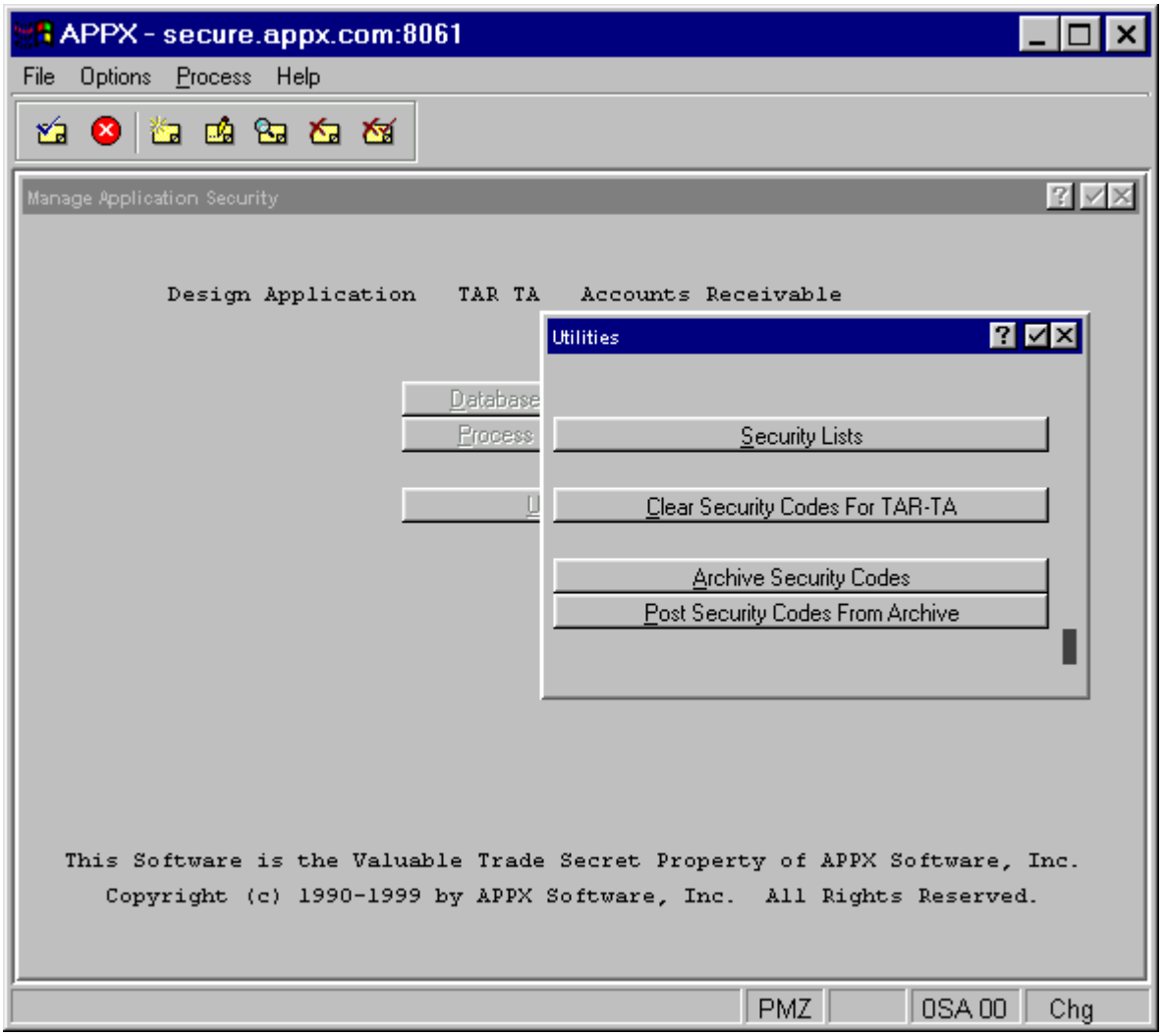


Figure 5-10. Security Management Utilities Overlay

The utilities available on this overlay are listed below:

- *Security Lists*
- *Clear Security Codes*
- *Archive Security Codes*
- *Post Security Codes from Archive*

These utilities are described in the following sections of this manual.

Security Lists

When you select the SECURITY LISTS option from the Security Management Utilities overlay, the Security Lists Item Selection overlay appears as shown in *Figure 5-11*.

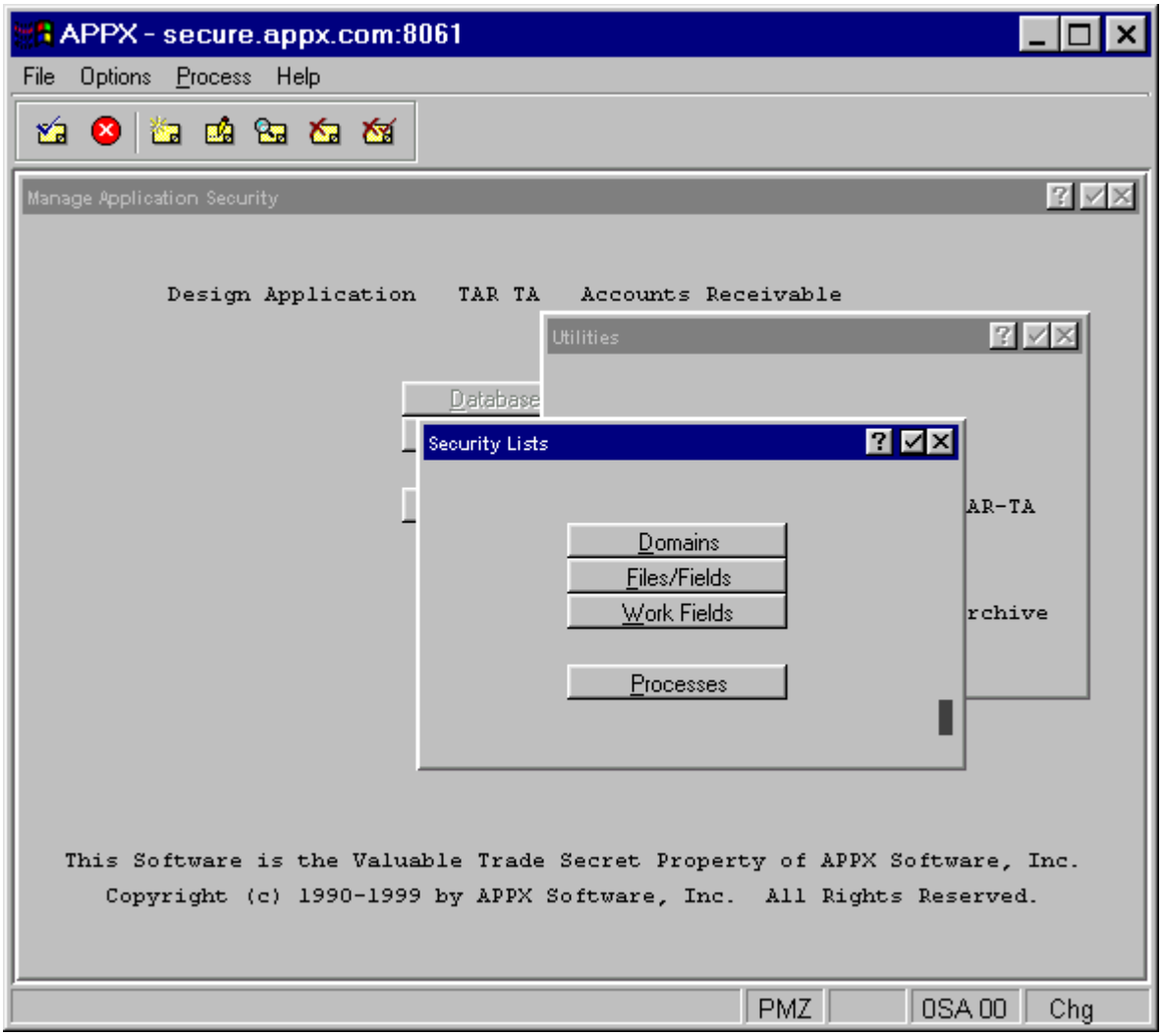


Figure 5-11. Security Lists Item Selection Overlay

This overlay allows you to print a list of all security codes for any type of item or process in your application. Once you have selected which security codes you wish to see, APPX will print a list of security codes either in hard copy or on the screen using the parameters you have defined.

Clear Security Codes

When you select the CLEAR SECURITY CODES option from the Security Management Utilities overlay, the Clear Security Codes overlay appears as shown in *Figure 5-12*.

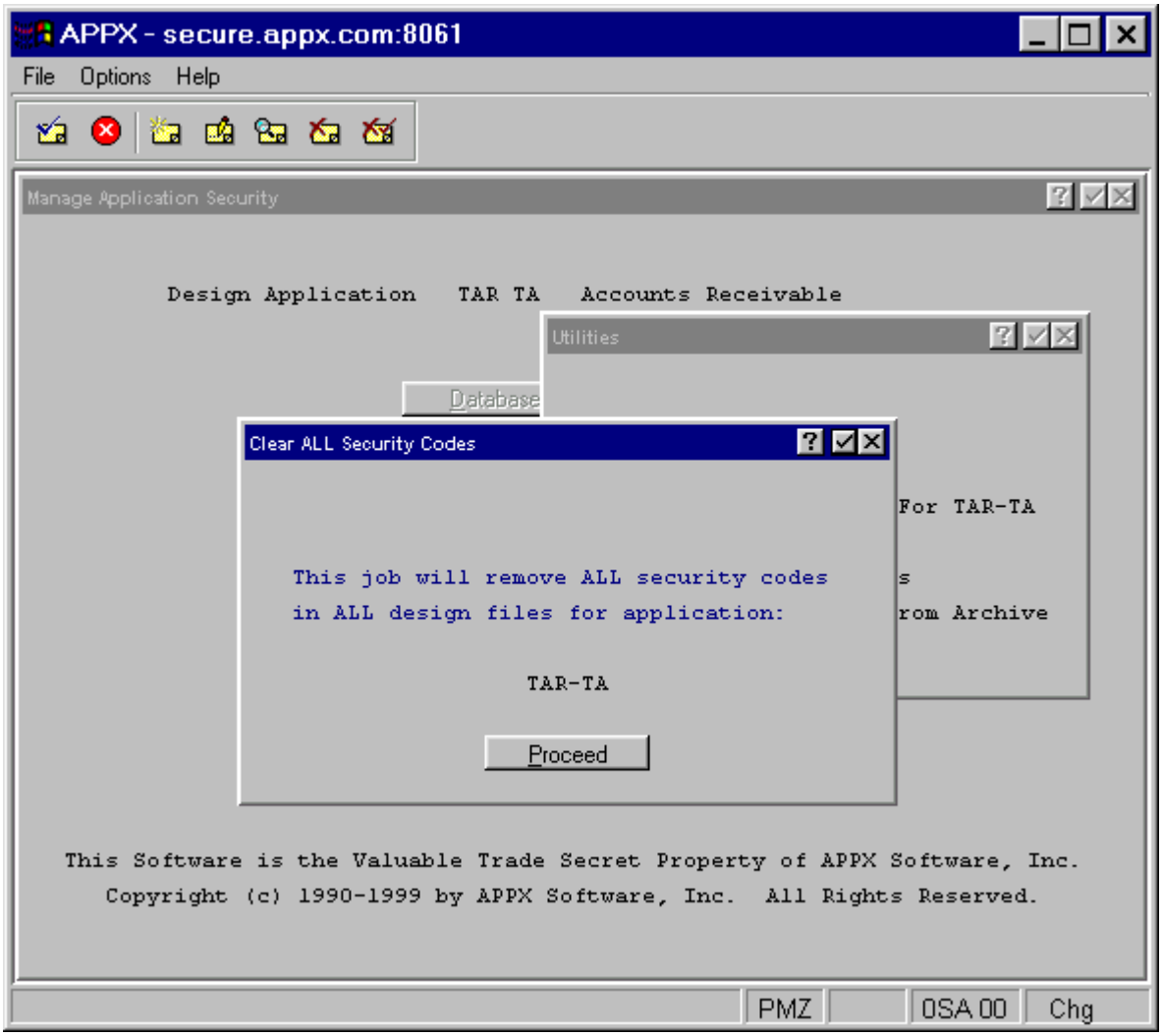


Figure 5-12. Clear Security Codes Overlay

This overlay provides for clearing all security codes in all design files for the currently selected application.

> Note that it is necessary to confirm that you want to continue by selecting the PROCEED option at the bottom of the overlay.

Archive Security Codes

When you select the ARCHIVE SECURITY CODES option from the Security Management Utilities overlay, the Archive Security Codes overlay appears as shown in *Figure 5-13*.

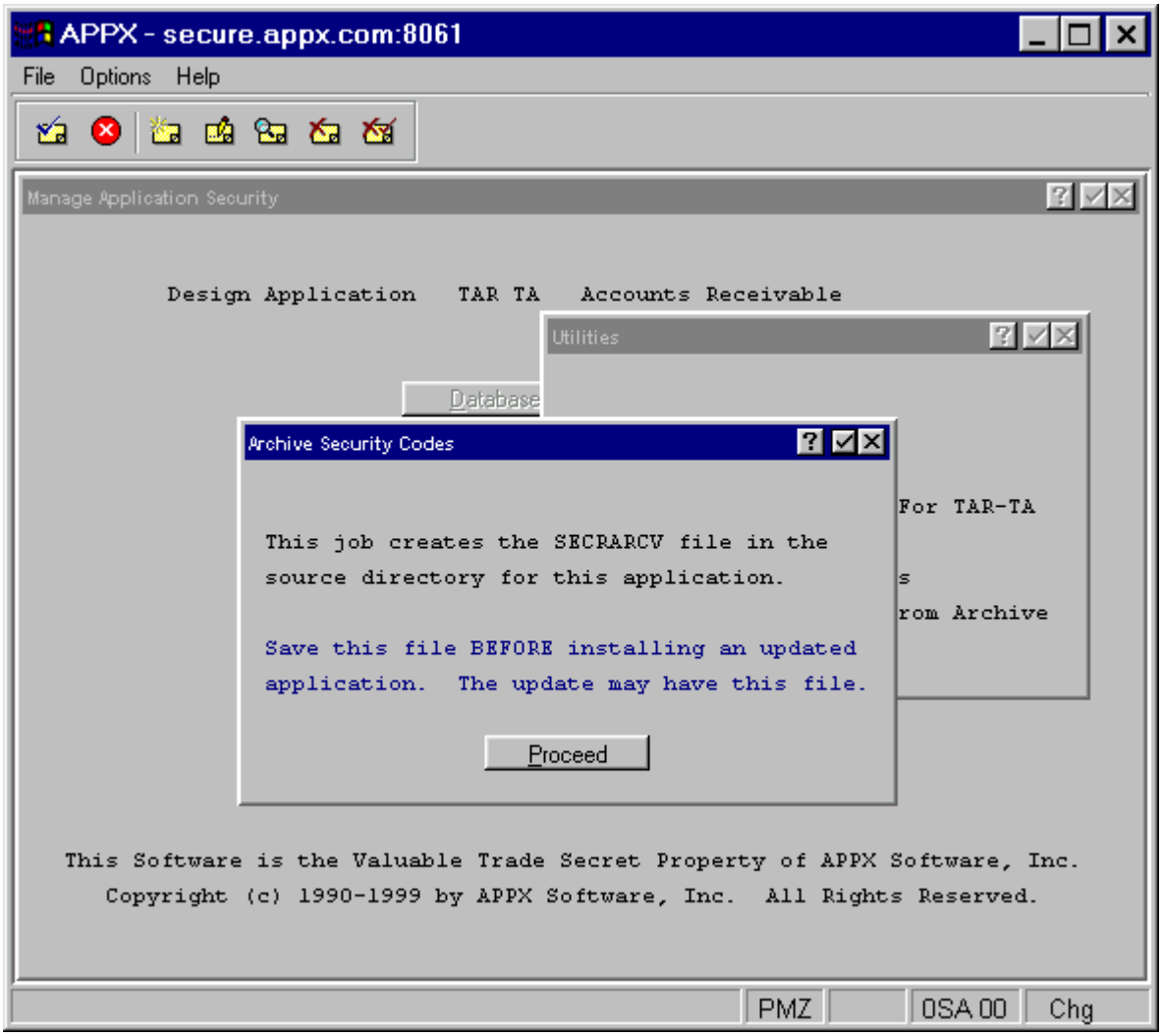


Figure 5-13. Archive Security Codes Overlay

This overlay provides for saving all security codes in the currently selected application by storing them in a central archive file called SECRARCV* (after first purging any existing archive records for this application from the file).

There are two reasons to archive your security codes. First, when you install an updated version of the application, you want to archive the security codes rather than recreating all of the security codes from scratch after the update. Second, the security codes in the archived file can provide a quick cross-reference of how security codes are being used.

> Note that it is necessary to confirm that you want to continue by selecting the PROCEED option at the bottom of the overlay.

***Note:** Contrary to what may be implied on the overlay, there is only one SECRARCV file. It is found in System Administration and serves all archiving functions whether security codes are archived for a single application through this utility or for multiple applications through the Archive Security Codes for All Apps utility found in Security within System Administration.

Post Security Codes from Archive

When you select the POST SECURITY CODES FROM ARCHIVE option from the Security Management Utilities overlay, another overlay appears as shown in *Figure 5-14*.

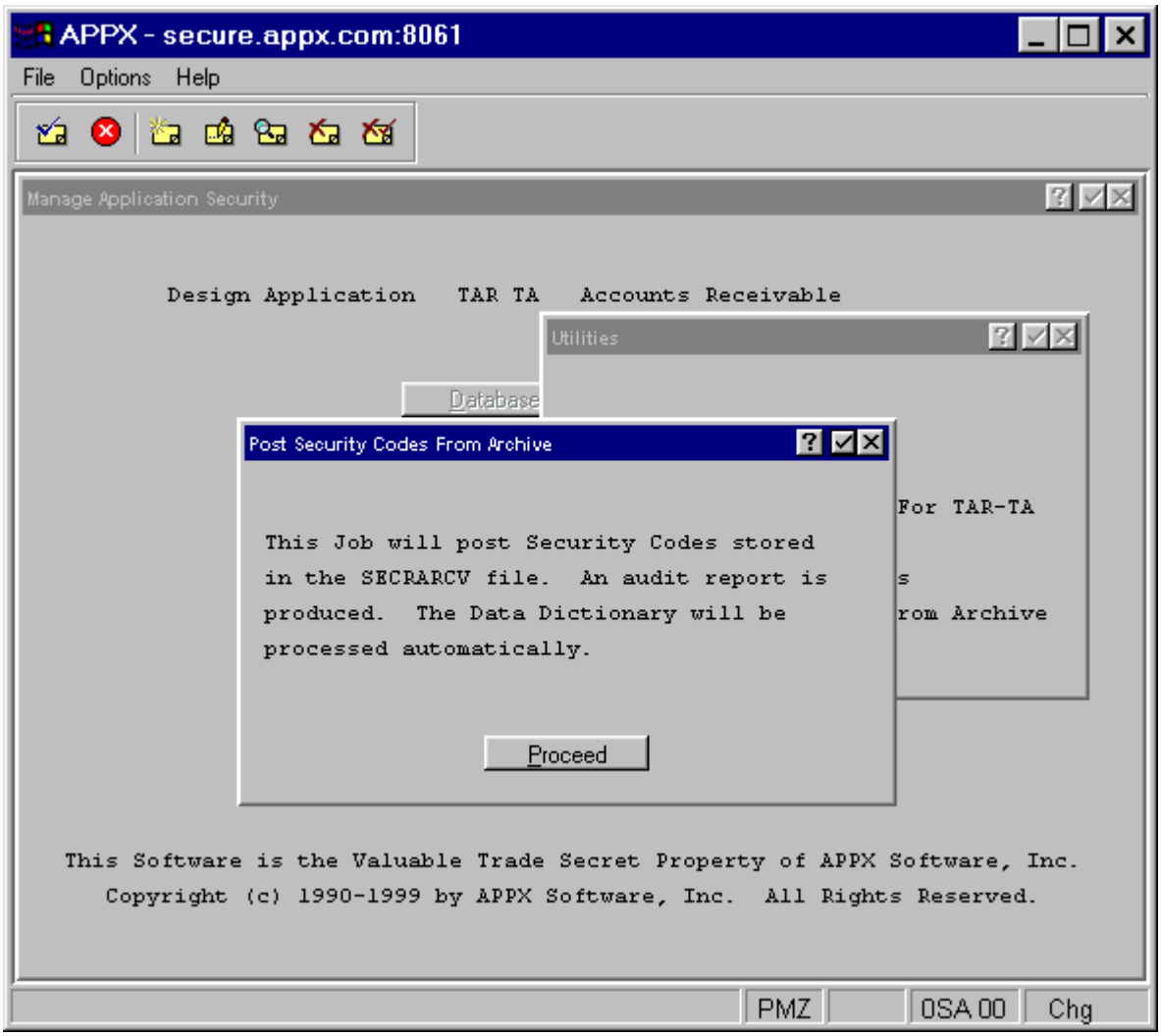


Figure 5-14. Post Security Codes from Archive Overlay

The POST SECURITY CODES FROM ARCHIVE option is normally used after installing an updated version of an application in order to retrieve the security codes stored in the archive file for that application and re-apply them. This assumes that, prior to installing the update, security codes were indeed archived for that application (by use of either the Archive Security Codes utility described in this chapter or the Archive Security Codes for All Apps utility found under Security within System Administration.

> Note that it is necessary to confirm that you want to continue by selecting the PROCEED option at the bottom of the overlay.

APPENDIX A

Printing in Unix

Overview - Printing in Unix

APPX generates print files into the "/tmp/" directory by default. You can change the location of the print files by setting the environment variable *APPX_PRT_FI_DIR* to the name of an existing directory. The directory specified by *APPX_PRT_FI_DIR* must include a trailing "/" character.

The interface to the Unix 'lp' command is in a shell script named 'appx_print'. 'appx_print' is a Bourne shell script invoked by APPX to deliver a print file to the Unix print spooler. The format of the options file is documented in 'appx_print' and is subject to change in future releases.

When APPX invokes 'appx_print', it passes two parameters to it: the name of an options file and the name of the print file. The options file is a flat ASCII file containing the Disposition information specified by the user at the time the print file was generated. The 'appx_print' script (as provided with APPX) simply generates an "lp" command line based on the disposition information contained in the options file.

'appx_print' creates a readable copy of its last call to 'lp' in "/tmp/appx_print.last". This file is useful for debugging.

This appendix includes a discussion on customizing 'appx_print' as well as descriptions of the disposition options used by 'appx_print'.

Customizing 'appx_print'

The 'appx_print' script can be replaced or modified if your spooling system provides additional functionality that the default script does not take advantage of. You may also wish to modify or replace the 'appx_print' script for debugging purposes. If you do modify 'appx_print', save the script under a new name and set the environment variable *APPX_PRT_SCRIPT* to the new script name. New or updated APPX releases will overlay the 'appx_print' script.

'appx_print' is expected to reside in the $APPXPATH directory. The name (but not the path) of 'appx_print' can be changed by setting the environment variable *APPX_PRT_SCRIPT*.

Underlines and Bold-Facing

APPX accomplishes both underlining and boldfacing using the same technique: overstriking using backspaces. In order to print an underlined character, APPX prints the desired character followed by a backspace character (hex 08), followed by an underline character. Boldfacing is accomplished by printing the desired character followed by a backspace followed by the desired character again. On printers that understand backspaces, this technique produces underlined and/or boldfacing. Some printers have configuration parameters that affect the interpretation of backspace characters.

If your printer does not treat backspaces as APPX expects, you can modify the ‘appx_print’ script to translate the backspace sequences into the form required by your printer.

Disposition Options

All disposition options are now available to the Unix user. However, only a subset of the disposition options are used by ‘appx_print’:

Print Mode

Spool

And ***Hold*** both pass the print file to ‘lp’ via ‘appx_print’.

Keep

Leaves the print file in the spool directory and does not invoke ‘appx_print’.

Printer Device Form Queue

The Printer “Device” and Form “Queue or Class Name” fields are used to specify a print queue and/or device name. If both fields are blank, the default printer (as defined by the “lp” command) is used. If one of the fields is non-blank, the other field is used as the printer name. If neither field is blank, the Form Queue and Printer Device are both sent to the “lp” command in the form “-dFormQueue:PrinterDevice”. Check your "lp" command to make sure it supports this option if you want to use it.

In summary, the following forms are used:

- Printer Device only is -dPrinterDevice
- Form Queue only is -dFormQueue
- Both Fields is -dFormQueue:PrinterDevice
- Neither Field means the default is chosen by lp

Printer Control/Form Control

The Printer Control and Form Control fields, if non-blank, are copied without modification to the “lp” command line. These fields can be use to pass “lp” command-line options such as those that change orientation or fonts. Any “lp” options that can be used at the Unix command line can be specified in either the Printer Control or Form Control fields.

APPENDIX B

Printing in Microsoft Windows

Overview - Printing in Microsoft Windows

A printer is identified on the disposition screen that may be presented to the user after any record selection or sorting options. That printer ID is a key to the Printers file in System Administration. The print queue subsequently used is determined by the printer defined by the System ID field in the Printers file for that Printer ID. This should exactly equal the name of a printer identified in Control Panel, Printers in Windows. This field is both case sensitive and spacing sensitive.

A program named 'winprint.exe' connects the print file generated by APPX with the Windows print spooler. 'winprint.exe' supports a number of keywords, printer and/or form control options, standard Windows form names, and certain paper sources. A description of these items is included in this appendix, along with a discussion of printer driver issues.

Customizing 'winprint.exe'

Under certain circumstances, you may want to customize 'winprint.exe'. In that case you can request the source code from APPX Software and make your own version. That version *should not* be named 'winprint.exe', because any subsequent upgrade would replace that executable with APPX's standard current version. Rather, call your new executable something else, store it in the *%APPXPATH%* directory, and set the environment variable *APPX_PRT_SCRIPT* to the new executable name.

Underlines and Bold-Facing

APPX accomplishes both underlining and boldfacing using the same technique: overstriking using backspaces. In order to print an underlined character, APPX prints the desired character followed by a backspace character (hex 08), followed by an underline character. Boldfacing is accomplished by printing the desired character followed by a backspace followed by the desired character again. On printers that understand backspaces, this technique produces underlining and/or boldfacing. Some printers have configuration parameters that affect the interpretation of backspace characters.

If your printer does not treat backspaces as APPX expects, you can modify 'winprint.exe' to translate the backspace sequences into the form required by your printer.

Keywords in 'winprint.exe'

Following is a list of keywords that are recognized by 'winprint.exe'. Most of the values for these keywords are retrieved from the disposition screen in APPX. This list can also be displayed by double clicking on *%APPXPATH%\winprint.exe*:

- **-banner** prints a banner page if present (not currently supported).
- **-config** is the name of the configuration file to use (the configuration file can provide additional options).
- **-copies** is the number of copies to print (not supported by some printer drivers).
- **-date** is the date/time to print report (not currently supported).
- **-disposition** is the disposition of the print file after printing. Valid choices are *SCRATCH* and *SAVE*; the default is *SAVE*.
- **-format** is the format of the print file (not currently supported).
- **-form_ctl** is the list of form-related options.
- **-form_id** is the Windows Form Name. See the list of valid form names, below.
- **-form_name** is the APPX Form Name (ignored).
- **-mode** is the print mode. Valid choices are *KEEP*, *HOLD*, and *SPOOL*; this parameter is currently ignored and defaults to *SPOOL*.
- **-notify** notifies the user when the report has completed printing, if present (not currently supported).
- **-priority** is the priority of the report (not currently supported).
- **-printer_ctl** is the list of printer-related options.
- **-printer_id** is the Windows Printer Name.
- **-printer_name** is the APPX Printer Name (ignored).
- **-user** is the User Name (ignored).
- **-log** provides the name of a debugging log file.

Printer/Form Control Options

These are options you can use in the ***Printer Options*** and ***Form Control*** fields in the Printers and Forms files in System Administration.

- **-orientation** is the report orientation; valid choices are =Portrait and =Landscape.
- **-collate** means that multiple copies of the report are collated, if present (not supported by most printer drivers).
- **-pitch** is the pitch to use in printing the report.
- **-font** is the name of the font to use in printing the report.

- **-ruler** prints a ruler on each page, if present.
- **-width** is the width of the form; use 'i' or 'm' as a suffix to indicate inches or millimeters (eg. 8.5i or 23m).
- **-length** is the length of the form; use 'i' or 'm' as a suffix to indicate inches or millimeters (eg. 8.5i or 23m).
- **-interactive** shows the printer setup dialog to the user if present; if "=none", any windows for network printing are suppressed.
- **-source** is the paper source (not supported by many printer drivers). See the list of valid paper sources, below.
- **-oem** translates characters to the OEM character set, if present.
- **-ansi** translates characters to the ANSI character set, if present.
- **-raw** passes the report to the printer without translation, allowing you to embed escape codes in your report file. If *–raw* is not used, the Windows print spooler filters out escape codes you may wish to pass to your printer.

Windows Form Names

Table B-1 set forth below contains a list of valid Windows form names along with their width and length dimensions. You can also create a new form with a custom width and length on Windows NT and 2000 systems by going to: Start; Settings; Printers; File, Server Properties, Create a New Form. Then, you can reference that form through the *–form_id* keyword.

The entry in the ***Form Control*** field in the Forms file in System Administration should be in the following format, using the exact text as specified:

-form_id=Windows_Form_Name

Table B-1. Windows Form Names

Windows Form Name	Width	Length
=”10 by 14 sheet”	10 in	14 in
=”11 by 17 sheet”	11 in	17 in
=A3	297 mm	420 mm
=A4	210 mm	297 mm
=”A4 small”	210 mm	297 mm
=A5	148 mm	210 mm
=B4	250 mm	354 mm
=B5	182 mm	257 mm
=C	17 in	22 in
=D	22 in	34 in

=”#10 envelope”	4.125 in	9.5 in
=”#11 envelope”	4.5 in	10.375 in
=”#12 envelope”	4.75 in	11 in
=”#14 envelope”	5 in	11.5 in
=”#9 envelope”	3.875 in	8.875 in
=”B4 envelope”	250 mm	353 mm
=”B5 envelope”	176 mm	250 mm
=”B6 envelope”	176 mm	125 mm
=”C3 envelope”	324 mm	458 mm
=”C4 envelope”	229 mm	324 mm
=”C5 envelope”	162 mm	229 mm
=”C6 envelope”	114 mm	162 mm
=”C65 envelope”	114 mm	229 mm
=”DL envelope”	110 mm	220 mm
=”Italy envelope”	110 mm	230 mm
=”Monarch envelope”	3.875 in	7.5 in
=”6 3/4 envelope”	3.625 in	6.5 in
=E	34 in	44 in
=Executive	7.25 in	10.5 in
=”German legal fanfold”	8.5 in	13 in
=”German standard fanfold”	8.5 in	12 in
=”U.S. standard fanfold”	14.825 in	11 in
=Folio	8.5 in	13 in
=Ledger	17 in	11 in
=Legal	8.5 in	14 in
=Letter	8.5 in	11 in
=”Letter small”	8.5 in	11 in
=Note	8.5 in	11 in
=Quarto	215 mm	275 mm
=Statement	5.5 in	8.5 in
=Tabloid	11 in	17 in

Paper Sources

A list of valid paper sources is set forth below. Remember that many printer drivers do not support the selection of paper sources. The entry in the ***Form Control*** field in the Forms file in System Administration should be in the following format, using the exact text as specified:

-source=Paper_Source

=Upper

=Lower

=Middle

=Manual

=Envelope

=”Manual Envelope”

=Automatic

=Tractor

=”Small format”

=”Large format”

=”Large capacity”

=”Paper cassette”

=”User defined”

Printer Driver Issues

Printer drivers have two ways to specify printer parameters: interactively, and through the manufacturer’s printer driver API. Normally, APPX uses the printer driver API. This is a mechanism for printer drivers to receive parameters that control printing from applications such as APPX. The printer driver then tells the printer how to print the passed document.

If you specify *-interactive* in your APPX ***Form Control***, you will access the printer driver’s interactive specifications editor. This is similar to the one you see in Microsoft Word when you click File, Printer Setup. This display is not available if you run APPX through Client/Server, only if you are running appx.exe directly, from the server or through a mapped drive.

Often the manufacturer’s printer driver API interface is not as complete as the printer manufacturer’s interactive interface. For example, some ignore *-form_id*, which is used to control form length.

The following technique may give you a way to print a custom form length without having to use the *-interactive* screen on your APPX server:

1. Go into Windows and create a new printer.
1. When it gets to 'Printer Manufacturer', select 'Generic', 'Text Only'.
1. Finish setting up the new printer.
1. Go back into Start; Settings; Printers, and edit the properties of this new printer.
1. Look at the 'Font Selection' tab. It allows you to enter hex codes for various pitches, Bold, and Underline. You may want to use a particular pitch for your custom form length.
1. Enter the control codes needed for your custom form length into one of these entries.
1. Define in APPX a ***Printer Control*** that passes a keyword that calls the pitch that you put your custom form control codes into.

The 'Printer Commands' tab also lets you send some hex codes to the printer.

If you want to pass control codes to your printer by any other method, you must use the *-raw* keyword in your ***Form Control***, and embed the control codes in your Output Image.

APPENDIX C

Command Line Options

Standard APPX Command Line Options

The following command line options are available in standard APPX, whether running APPX from Unix or from Windows/NT:

```
-k define a new keymap
-m=<[user_id:]keymap_name> override keymap ID
-d=<database_id> override startup database
-a=<application_id> override startup application
-t=<process_type> override startup process type
-p=<process_name> override startup process name
-i=<flat_file_name> convert a flat file into APPXIO  format
-e=<data_file_name> convert an APPXIO file into  a flat
                              format
-v=<data_file_name> verify the integrity of an  APPXIO file
-an=<data_file_name> analyze an APPXIO file
-l=<log_file_name> start a license server
```

Invoking in Unix

In Unix, invoke command line options by running APPX with the following command structure:

```
appx  -(x)
```

> Corrected: the printed manual shows a dash (–) where the command takes a hyphen (-).

where (x) is the option of choice.

Invoking in Windows

In Windows, invoke command line options by running APPX with the following command structure:

```
appxutil.exe  -(x)
```

> Corrected: the printed manual shows a dash (–) where the command takes a hyphen (-).

where (x) is the option of choice.

Restrictions

An application_id and process_type must always be included when specifying a process_name. If an application_id is included without specifying a process_type/name, the startup menu for the application is used.

If no application_id is specified, the startup application and menu for the database is used. If no database_id is specified, the startup database defined for the user within System Administration is used. If the startup database is also unspecified, the APPX main menu is used.

The valid process_types are Menu, Job, Input, Output, Update, Action, Inquiry, Query, Status, and Subroutine. Use underscores (_) in place of any embedded spaces within parameters.

Option Detail

Option detail is provided as needed for some command line options below:

`-k`

Invokes the keyboard mapper to either replace an existing keyboard map or to define a new keyboard map. The keymap will be stored by Terminal Type and APPX User ID so that you only have to define a desired map once for each terminal you will be using. Note that if you run APPX using the APPX Client, the keymap is hard-coded and may not be changed.

`-m=[user_id:]keymap_name`

Overrides the default name for this keymap. For example, if you are using a vt320 terminal your map will be called vt320 by default. This option is particularly useful if, for example, you have a laptop computer that has a software package to emulate a vt320 terminal and you dial in to your system remotely. Although the terminal setting will appear the same to the system, you can define a different keyboard map (called perhaps "laptop") to use when you log in remotely.

`-i=filename`

Converts an APPX flat ASCII portable file into an APPXIO format portable file. APPX portable files must be converted to and from flat ASCII format when moving between systems of different byte ordering.

`-e=filename`

Converts an APPXIO format portable file into an APPX flat ASCII portable file. APPX portable files must be converted to and from flat ASCII format when moving between systems of different byte ordering.

`-v=filename`

Checks the integrity of an APPXIO format file. "filename" specifies the file to be analyzed. This does the same thing as interactive “Verify Files Menu", "Verify Integrity" from the Database, Design File, and System Administration File Management menus, but from a command line.

If the invocation of this command results in no output, then the file passes verification. Any output from this command means that the file fails verification.

`-an=filename`

Performs an interactive analysis of an APPXIO format file. "filename" specifies the file to be analyzed. Type "help" at the prompt to get a list of valid analysis commands. Valid commands are:

- **verify key #all** counts the number of records in the .dat portion of the file, verifying the data as it goes, and verifies all keys in the .key portion of the file.
- **help** lists all available commands.
- **dump header** gives you information about the file, including the number of keys.
- **count rec** gives you a count of the number of records that *filename.dat* believes it contains.
- **verify key #0** verifies the primary index, and gives you the number of records.
- **verify key #1** verifies the first alternate index, and gives you the number of records.
- **verify key #2** verifies the second alternate index, and gives you the number of records. You can continue on this way until all index files are verified.
- **quit** exits the "-an=" utility.
- **verify freelist** gives you an error message if the file's freelist is damaged. The freelist is the list of pointers in APPXIO files that point to deleted records. If this list gets damaged, it may point to a live record, indicating erroneously that it is deleted. **verify key #all** does not detect damaged freelists. If you get an error message from the **verify freelist** command, run **verify key #all**; if no errors occur, you can safely export and re-import the data to initialize the freelist.

If the file does not pass **#all** verification, but does pass **count rec** verification, it means that the *.dat portion of the file is probably undamaged, but the *.key indices are probably damaged. In this case, rename or delete the *.key file, then run *Create Files* in Database Management. This will recreate your indices from scratch. Then, if the freelist was damaged, you can safely export and re-import the data to initialize the freelist.

If the file does not pass **count rec** analysis, you have a seriously damaged *.dat file. At this point you should either run the '*recover.exe*' procedure, or if necessary restore the file from a backup.

APPX Client Command Line Options

The following command line options can be appended to the `COMMAND LINE=` line of the `login.ini` file, located in the primary destination directory of the APPX Client. (The default location is `C:\appx\java\login.ini`)

`-mode=(local|remote|both)`

Controls which login panels are shown. This also determines which type of login to do if the login dialog box is skipped. The default is "both".

`-pathname={path & filename of the local appx.exe file}`

Defaults the location and name of the APPX server executable when running APPX locally. This is also used when the login dialog box is skipped and "-mode" is set to "=local".

`-port={server port number}`

Defaults the remote server port number on the login dialog box. This is also used when doing a remote login when the login dialog box is skipped. The default is "8060".

`-host={server hostname}`

Defaults the remote host field on the remote login pane of the login dialog box. This is also used when the login dialog box is skipped.

`-user={server OS user ID}`

Defaults the user ID on the remote login pane of the login dialog box. This is also used when the login box is skipped.

`-password={server OS password}`

Defaults the password field on the remote pane of the login dialog box. This is also used when the login dialog box is skipped.

`-look=(java|windows|motif)`

Defaults the "look and feel" setting on the Options pane of the login dialog box. This setting is also used when the login dialog box is skipped. The default is "Windows".

`-database={APPX server Database ID}`

Instructs the server to start up in a particular database.

`-application={APPX server Application ID}`

Instructs the server to start up in a particular application.

`-ptype=(menu|job|input|output|...etc...)`

Instructs the server to use this as the startup process type.

`-process={APPX server process name}`

Instructs the APPX server to use this as the startup process name. If the database, application, process type, and process name are all specified, the Client will run the specified process in the specified application and database when the system starts up. If the process type and name are not specified, the Client will start up in the specified database and display the specified application's main menu. If the

application is not specified, the Client will start up in the specified database and display the menu identified in the Databases file in System Administration. If the database is not specified, APPX displays its main menu.

-fg=(black|blue|cyan|darkGray|gray|green|lightGray| magenta|orange|pink|red|white|yellow|#rrggbb)

Sets the default text foreground color. You can select one of the predefined colors, or define your own with the “#rrggbb” setting, where you replace “rr” with the amount of RED in hexadecimal, and the same with the “gg” (GREEN) and “bb” (BLUE) values. The default is black.

-bg=(black|blue|cyan|...etc...)

Sets the default text background color. See “-fg” for more information.

-boldFg=(black|blue|cyan|...etc...)

Sets the default text foreground color for bold text. See “-fg” for more information.

-editFg=(black|blue|cyan|...etc...)

Sets the default text foreground color for edit fields. See “-fg” for more information.

-rows={window size in rows}

Establishes the initial screen size in rows. The default is 21.

-cols={window size in columns}

Establishes the initial screen size in columns. The default is 80.

-bmode=(tile|expand)

Defines how background tiles are painted in the Client window. The “tile” setting uses the tile image over and over again to fill the window. The “expand” setting resizes the image to just fill the space inside the window. This is used with the “-tile#=” setting. The default is “tile”.

-splash={url to an image}

Establishes a background image in the main client window during the login process. The way it is painted is determined by the “-smode=” setting.

-smode=(tile|expand)

Defines how the splash image is painted in the Client window. The “tile” setting uses the tile image over and over again to fill the window. The "expand" setting resizes the image to just fill the space inside the window. This is used with the “-splash=” setting. The default is “expand”.

`-tile#={url to an image}`

Defines an array of images that the client can use to fill window backgrounds. “-tile1=” defines the base wallpaper image. “-tile2=” defines the wallpaper for all windows one up from the base window. “-tile3=” sets the pattern for all windows two up from the base window, etc. You can define as many or as few as you like. The Client will cycle back to the first image if you have more windows layered on the screen than it has tile images for. This is used with the “-bmode=” setting.

`-shell={command line to start a document}`

Overrides the default command line used to “start” a document. Starting a document tells Windows to open a document and lets Windows use the association to decide what program to use. This defaults to “start {document}”, which may not work on NT. NT may require this to be set to start a command shell like “cmd.exe /c start”.

`-font=fixed`

Turns off the proportional font normally used by the Client. If you are having trouble lining up text on your application screens, try this setting. Fixed fonts set aside the same space for each letter or number, and are therefore easier to align.

`-optkey={character}`

Resets the key used as the Option key. The default is the single-back-quote “`” character. To set the option key to the “~” character, use “-optkey=~”.

`-title="whatever you want here"`

Sets the window title to the quoted text. Special data can also be inserted into the title by using any of the following tags: $(host), $(user), $(pathname), $(pid), and $(port). For example, you might use: -title="Steve's APPX Client, server PID is $(pid)"

`-hideMenu`

Hides the Java/Client Menu bar.

`-hideToolbar`

Hides the Java/Client Toolbar.

APPENDIX D

Creating a Custom APPX Engine

Linking External 'C' Programs into APPX

The ability to invoke external programs and routines written in ‘C’ is available from within APPX. Once the code has been compiled, it can be called anywhere within APPX ILF code with the “CALL” statement.

Due to the method of linking on the IBM RS6000 machines, any externally executable program can be called directly from the ILF code without having to compile and link it into the APPX engine. On all other platforms, to make the required code executable from within APPX, we have provided a stub program named ‘call.c’ in which you can include your required code. This program is included in the “linkable” subdirectory created below the directory into which you installed APPX.

To call a routine from the ‘call.c’ program, put the code you want to call within the program and make entries in the places indicated. Refer to the examples *hello* and *get_addr* included with your installation materials.

Then, compile the code and link it to the engine by issuing the following command:

```
make appx -f Makefile.callable
```

This will compile the file ‘call.c’ and then link the executable code into the APPX engine.

To call the required routine, for example *hello*, you insert a CALL statement within the ILF code where you want the routine to be called. In the case of the *hello* routine, the line would be:

```
CALL .hello RESIDENT? N END? N FAIL 0
```

Another example is included called *get_addr* which will return the address of the first parameter passed to it as the second parameter. The necessary PASS and CALL statements to execute this routine are:

```
TRAP
PASS --- TEMP 79 FIELD SHARE? Y
PASS --- AI FIELD SHARE? Y
CALL .get_addr RESIDENT? N END? N FAIL? 0
SET --- AI = --- AI
```

Corrected: the printed manual shows long dashes in these lines; in ILF the current application is written as three hyphens (---).

This will show that AI, initially zero, will contain the address of TEMP 79 upon returning from *get_addr*.

Note that the leading period in the CALL statement for the function is required.

Linking External File Mgt Systems into APPX

APPX allows you to link an external file manager into the engine to replace or supplement its own file manager. The currently supported external file managers are Vision and C-ISAM on Unix or Unix-related systems only.

The files required to link an external file manager into APPX are now included with the installation materials. These files can be found in the “linkable” subdirectory created below the directory where you installed APPX. To link an external file manager into APPX, change directories to the *linkable* directory:

cd $APPXPATH/linkable

Now you must research the locations of your linkable archive files for either Vision or C-ISAM. These files must be obtained from ACUCOBOL or Informix, and may vary depending on the operating system you have. They may also vary depending on the version of Vision or C-ISAM you have. For Vision the filename is usually ‘runcbl.a’. For C-ISAM the filename is usually ‘libisam.a’. Specify the locations of these files when entering the command to create the new engine. In the examples below the parameters ‘cisam=...’ and ‘vision=...’ specify the locations of C-ISAM and Vision archive files respectively. You may enter zero, one or both of these parameters. Any parameter that is not entered will not be available in the new engine. Examples:

To create an engine with C-ISAM but no Vision:

```
make -f Makefile.relink appx cisam=location/libisam.a
```

To create an engine with Vision but no C-ISAM:

```
make -f Makefile.relink appx vision=location/runcbl.a
```

To create an engine with both interfaces:

```
make -f Makefile.relink appx vision=location/runcbl.a  cisam=location/libisam.a
```

To create an engine with neither optional interface:

```
make -f Makefile.relink appx
```

When the make is done you should have a new APPX engine, which you will use in place of the one shipped with the standard release. To test the new engine, execute APPX without replacing the old engine with the following command:

```
./appx
```

To make a file use the Vision or C-ISAM interface you need to remove the existing file. If the file contains data you must export it first. Use Database Management for the appropriate database and application, and select the 'Export' option. Then scratch the existing file using the 'Scratch' option. Now you must indicate to APPX the appropriate file management system (FMS) to use. The specification is entered using the 'File Specifications' option. The column labeled 'FMS' must be changed to the correct value. The values are:

- '1' = APPXIO (standard APPX file manager)
- '2' = Vision
- '3' = C-ISAM
- '4' = Sybase
- '5' = Oracle
- '6' = ODBC
- '7' = APPX/Net

You can select any FMS for any data file. Design and System Administration files **must** use FMS 1.

Once you have selected the correct FMS you will need to create the file by using the 'Create' option. If you receive an error you either selected the incorrect FMS value or mis-linked your engine. Contact your support center for assistance. If you exported old data you may now import it by using the 'Import' option in Database Management. If you are going to access this data with ACUCOBOL read the notes about *APPX_PD_SIGNS* below **before** importing your data.

If the engine functions correctly you should replace the standard APPX engine with your customized one. You should either backup the standard engine or rename it. The command to rename it is:

```
cd ..
mv appx appx.standard
```

Then copy in the new engine:

```
mv linkable/appx .
```

Change ownership and permissions:

```
chmod u+s appx
chown appx appx
```

You may now delete the linkable directory if you desire:

```
rm -r linkable
```

Your customized APPX is now installed.

APPX references the environment variable *APPX_PD_SIGNS* to determine what values should be used for the sign portion of a packed decimal number. By default APPX uses 'C' to indicate positive numbers and 'D' to indicate negative numbers. ACUCOBOL uses different values. If you are going to be exchanging data between APPX and ACUCOBOL programs you **must** set the value of *APPX_PD_SIGNS* to FD using the following line in your profile or APPX startup script:

```
export APPX_PD_SIGNS=FD
```

The first letter of *APPX_PD_SIGNS* is the value to use for positive numbers and the second letter is the value for negative numbers. NOTE: if you exported any data during the steps above **and** you are using ACUCOBOL (or some other language which uses different sign indicators), you must re-import your data with *APPX_PD_SIGNS* set correctly. The signs **must** be stored the same for **all** data accessed by APPX for a particular installation. If some of your data is stored in different FMS systems, you **must** export and re-import **all** of your data whenever changing the value of *APPX_PD_SIGNS*, or arithmetic comparisons and calculations may not work correctly.

Linking an RDBMS into APPX

On Unix systems, there are pre-linked versions of APPX for Oracle, Sybase, and DB2. They can be found either in the *{appx engine location}* directory or the *{appx engine loction}/linkable* directory as 'appx.oracle', 'appx.sybase', and 'appx.db2'. (On Windows/NT systems, if the requested RDBMS has been properly installed, APPX will find all the '.dll' files it needs in the appropriate place at the appropriate time.) If you need to interact with both Oracle and Sybase you will have to make a custom-linked APPX engine, and you should contact APPX Software to find out how to accomplish this. The information needed varies by operating system, and may also be different for different versions of Oracle, Sybase, and DB2.

Table D-1 below shows the relationships between the supported operating systems and the various database managers. Note that Vision and C-ISAM are not supported on Windows/NT systems, and the ODBC interface is only supported on Windows/NT.

Table D-1: APPX Platform/Database Support

(As of APPX release 4.1.3)	Vision	C-ISAM	Oracle	Sybase	DB2	SQL Server	MS Access
HP-UX	Manual	Manual	Pre-linked	Pre-linked	Pre-linked		
AIX	Manual	Manual	Pre-linked	Pre-linked	Pre-linked		
Solaris	Manual	Manual	Pre-linked	Pre-linked	Pre-linked		
Linux	Manual	Manual	Pre-linked	Pre-linked	Pre-linked		
Windows/NT			Pre-linked	Pre-linked	Pre-linked	ODBC	ODBC

Manual: The installer must manually link Vision or C-ISAM into APPX.

Pre-linked:	Interfaces are pre-linked into APPX; for multiple links a custom engine is required.
ODBC:	Data can be accessed through the APPX/ODBC interface.

APPENDIX E

APPX Environment Variables

Overview - APPX Environment Variables

APPX environment variables are used to modify the behavior of APPX at runtime. Behavior can be modified for all sessions of APPX for all users, or for a specific user or users only.

A list of all APPX environment variables is found in *Table E-1*. Following this table are three sections containing more detailed information regarding the usage and setting of these variables. In each of these major sections, *General Environment Variables, Database (RDBMS) Interface Variables,* and *Debugging Variables*, environment variables are grouped alphabetically within category based on the feature of APPX being controlled by the variable.

The final section of this Appendix, *Setting Environment Variables*, explains where these variables can be set.

Table E-1: APPX Environment Variables

Table E-1. APPX Environment Variables

Environment Variable	Usage
APPX_0SA_PATH	Tells a PC network node to look to a server's disk drive for its Sys Admin files
APPX_184_IPC	Reverts to pre-2.0 IPC semantics for RELATED processes
APPX_184_MASKS	Reverts to version 1.8.4 (and earlier) numeric display mask interpretation
APPX_200_ALIGNMENT	Reverts to version 2.0 (and earlier) numeric alignment
APPX_ACCESS_ALL_FILES	Reverts to pre-3.2 behavior when opening files
APPX_ALLOW_MISALIGNS	Converts Numeric Binary Alignment ERROR messages into WARNINGs
APPX_BG_SCRIPT	Uses a different script to submit batch jobs marked as “Separate Task? = Y”
APPX_BIG_BUTTONS	Specifies that the “big button” GUI style should be used
APPX_CGIDATA_LOG	File name to log CGI data parsing debugging information
APPX_CHECK_MEM	Causes APPX to periodically pause and verify its allocated memory
APPX_CISAM_DEBUG	Activates CISAM debug options with detailed error message reporting
APPX_DATA_SERVER	Hostname and port to which APPX/Net or APPX/ODBC will connect
APPX_DB_TRAPS	Allows Ctrl-C stepping into ILF debugger outside of Application Design
APPX_DBG_CODE	Debug level for APPX_SQL_CMD; almost always set to 0xFFFFFFFF
APPX_DBG_VFY	File name to log EM rebuild checking results
APPX_DISABLE_DYNAKEYS	Disables the ability of end users to create Dynamic Keys to data files
APPX_DISABLE_TCP_NODELAY	Disables the use of the TCP “NODELAY” option

Environment Variable	Usage
APPX_DYNA_LOG	File name to log Dynamic Key generation debugging information
APPX_DYNAKEY_PROMPT	Sets a threshold of records processed during Dynamic Key creation
APPX_END_PARAGRAPH	Tells APPX to use hex(0A) as an end-of-paragraph character within text fields instead of hex(B6)
APPX_FAKE_TIME	Causes APPX to think the current date/time is other than the OS date/time
APPX_FD_CACHE_CODE	Determines how much APPX_FD_CACHE_LOG logging to perform
APPX_FD_CACHE_LOG	File name to log all file opens and their success or failure status
APPX_FMT_IO_LOG	Readable form of IO_LOG
APPX_HOST	Presentation Client under Unix: Specify Presentation Server hostname
APPX_IO_LOG	File name where all I/O is logged
APPX_ISKEYSEG	Overrides APPX normal behavior of consolidating multiple key segments
APPX_KEATERM	Select whether a set of “smart buttons” for APPX will be displayed by KEA
APPX_KEYMAP	Sets the name of the default keymap to load; same as using the -m= option
APPX_LS_DBG	Specifies the debug level when debugging license server issues
APPX_LS_DELAY	Number of seconds to wait before License Server validates its license info
APPX_LS_GHOSTS	Causes APPX to revert to clearing out “ghost” sessions from the USAGE file
APPX_LS_INTERVAL	Number of minutes between checks of USAGE file information
APPX_LS_LOG	Denotes where to log license server client debug information
APPX_MSG	Centers the value of this variable on the bottom line of the screen
APPX_NET_FEATURE_MASK	Allows you to hard code a list of supported client/server features
APPX_NET_HASP	Specifies a network security HASP to login to
APPX_NET_HEADERS	Enables APPX client/server network packet headers for debugging
APPX_NET_LOG	File name to log Presentation Client/Server debugging information
APPX_NET_LOG_MASK	Debug level for APPX_NET_LOG, almost always set to 0xFFFFFFFF
APPX_NETIO_CODE	Debug level for APPX_NETIO_LOG, almost always set to 0xFFFFFFFF
APPX_NETIO_LOG	File name to log APPX/Net client side debugging information
APPX_NETIO_SERVER	File name to log APPX/NET server side debugging information
APPX_NO_END_CONSTRAINTS	Disables performance optimization to use constraints on upper range limits
APPX_NO_NEG_INT	Reverts to pre-3.3 semantics when assigning a negative number to signed binary
APPX_NO_STACK_DUMP	Disables stack dumps on Windows systems
APPX_ODBC_AUTOCOMMIT	Enables the auto-commit flag for SQL Server
APPX_ODBC_CODE	Debug level for APPX_ODBC_LOG, almost always set to 0xFFFFFFFF
APPX_ODBC_ISOLATION	Requests transaction isolation level 0 from the ODBC database being accessed
APPX_ODBC_LIB	Specifies the ODBC driver manager DLL location for dynamic linking

Environment Variable	Usage
APPX_ODBC_LOG	File name to log APPX/ODBC debugging information
APPX_OLD_PD_MUL	Disables fixes to packed decimal multiplication made to APPX version 3.0
APPX_ORACLE_CONNECT	Specifies what user ID and password to use to login to Oracle
APPX_ORACLE_FIRST_ROWS	“Hints” to Oracle that you want it to optimize its query processing
APPX_ORACLE_LIB	Specifies the Oracle library location for dynamic linking
APPX_ORACLE_PARAM_INSERTS	Use parameterized INSERTs to insert data into Oracle tables
APPX_ORIO_LIBS	Specifies Oracle libraries when dynamically linking from APPX
APPX_PD_SIGNS	Tells APPX what byte pattern to use for the sign bits of packed decimals
APPX_PIVOT_ITEMS	Used for Y2K compatibility
APPX_PIVOT_YEAR	Used for Y2K compatibility
APPX_PORT	Presentation Client under Unix: Presentation Server port number
APPX_PORTABLE_DATE	Ignores the "bad portable date" error when importing SPEED II files
APPX_PREFIX	Specifies the drive prefix of the APPX installation on NT for UniQue spooler
APPX_PRELOAD_MAX	Avoids a “panic-memory allocation error” when loading APPX across networks
APPX_PRT_FI_DIR	Sets the disk location where print files are created and stored
APPX_PRT_FI_SIZE	Determines how much space to pre-allocate for a print file (VMS only)
APPX_PRT_SCRIPT	Defines a different script for print jobs
APPX_PRT_SCRIPT_HIDE	Hides printer pop-up window when printing
APPX_PSWD	Presentation Client under Unix: Specify password
APPX_RDBMS_NDK	Chooses how APPX creates RDBMS date key segments
APPX_REMAP	Presentation Client under Unix: Invoke appxkeys remapping
APPX_RUN_MEM	Overrides the amount of memory APPX will use for an in-memory query
APPX_SCAN_PROC	Uses an alternate SCAN process; superseded by ‘Hook’ records
APPX_SCRIPT_IN	“Plays back” all keystrokes recorded in it as if user had manually keyed them
APPX_SCRIPT_OUT	Records all user keystrokes (and selected Processes) into an output file
APPX_SCRIPT_STEP	Causes APPX to pause before each APPX OPTION key is sent from script
APPX_SCRIPT_STOP	Causes APPX to exit when a recorded screen differs from the actual screen
APPX_SERVER	Location of APPX Presentation Server engine
APPX_SHOW_PROGRESS	Activates status message during a QUERY, OUTPUT and UPDATE
APPX_SIGNAL_LOG	File name to log all error signals (like bus errors and segmentation faults)
APPX_SQL_BUFFER	Specifies size of SQL command buffer when communicating with an RDBMS
APPX_SQL_CMD	File name of ASCII log file for SQL backend I/O traffic
APPX_SRVR_TIMEOUT	Specifies the interval in minutes between “is the other system alive?” pings

Environment Variable	Usage
APPX_SS_CODE	Debug level for APPX_SS_LOG, almost always set to 0xFFFFFFFF
APPX_SS_LOG	File name to log HASP initialization and verification information
APPX_STD_PGH_MASK	Sets the date mask for dates displayed in page headings
APPX_SUPPRESS_UID_SWITCH	Tells APPX to leave all print files owned by “appx”
APPX_SYBASE_DATABASE	Default Sybase database name
APPX_SYBASE_LEVEL0	Enables Level 0 table locking in Sybase
APPX_SYBASE LIB	Specifies the Sybase library location for dynamic linking
APPX_SYBASE_PACK_SIZE	Packet size for Sybase data server communications
APPX_SYBASE_SERVER	Default Sybase server name
APPX_SYS_ERR_LOG	File name to log all system errors
APPX_TRAILER_COUNT	Tells APPX to append additional trailer bytes to each memory allocation
APPX_TRAP_RESTRUCT	Inserts TRAP statements in Restructure event points for debugging
APPX_TST_DIR	Points to a directory to capture screen images
APPX_UID	User ID and Password used by APPX/Net or APPX/ODBC upon connection
APPX_UNIXIO_PROCS	Causes APPX to log the process names as they are executed
APPX_UNIXIO_STATS	File name to log every APPXIO file open, close and I/O
APPX_USER	Presentation Client under Unix: Specify User ID
APPX_VERIFY_IO	Turns APPXIO I/O verification on or off
APPX_VFY_RESTRUCT	Forces restructure to occur even if APPX thinks one is unnecessary
APPX_VISION_LOCKS_PER_FILE	Maximum number of locks per file, per process, in Vision file system
APPX_VISION_MAXFILES	Maximum number of open files in Vision file system
APPX_VISION_MAXLOCKS	Maximum number of record locks per process in Vision file system
APPX_XLATE_LOG	File name to log language translation debugging information
APPX_ZERO_MALLAC	Workaround for a bug which was fixed in version 3.0
APPXPATH	Default location of APPX Sys Admin, Applications and Database files
HASPTTY	HASP Port Number for Linux and SCO Unix boxes
VISION_VLOG	Activates VISION debug logging into the fixed filename "v.log"
WINPRINT_DEBUG	File name to log winprint.exe debugging information

GENERAL ENVIRONMENT VARIABLES

APPX/Net Client and APPX/ODBC Variables

APPX/Net Client and APPX/ODBC Variables

APPX_DATA_SERVER

Specifies the hostname:port to which your client will connect to retrieve APPX data. It should be set to your APPX server's machine name followed by a colon ":" character, followed by a port number such as 8060. The hostname and port specified must correspond to a running WinAppxD or appxd (you can test this by trying to login to the desired hostname and port with "appx –c" or the Java client). This variable is almost always required when using APPX/ODBC on a PC to access Unix data.

Valid values: hostname:port

Default value: not set

APPX_UID

Specifies the server user ID and password to be used for authentication of your APPX/Net connection. The user ID and password must exist and be non-expired for login to be successful. This variable is almost always required when using APPX/ODBC on a PC to access Unix data.

Valid values: server_userid:server_password

Default value: not set

Backward Compatibility Variables

APPX_184_IPC

Causes APPX to revert to pre-2.0 IPC semantics for RELATED processes if set to 1.

Valid values: 1

Default value: not set

APPX_184_MASKS

Causes APPX to revert to version 1.8.4 (and earlier) numeric mask interpretation if set to 1. Otherwise, APPX uses the new default numeric mask interpretation in APPX 2.0 and later.

Valid values: 1

Default value: not set

APPX_200_ALIGNMENT

Causes APPX to revert to version 2.0 (and earlier) alignment for numeric fields if set to 1. Otherwise, APPX uses the new default numeric alignment in APPX 2.2 and later.

Valid values: 1

Default value: not set

APPX_ACCESS_ALL_FILES

Causes APPX to revert to pre-3.2 file opening behavior if set to 1. In APPX 3.2, a performance optimization was made to cause APPX to open a file only if it believes there is a chance of I/O actually being performed with that file. Prior to release 3.2 APPX would open any file that was referred to by a process.

For example, in APPX 3.1, the statement:

SET TAR CUSTOMER BALANCE = 0

caused APPX to open the CUSTOMER file even though no IO took place.

Also, referring to a file in a child constraint caused APPX to open the file when the parent was executed.

Now, in 3.2 and later, APPX opens a file only if it detects one or more of the following reference types:

- The file is the PCF for the process
- READ, READNEXT, or BEG/END READ statement is present
- WRITE, REWRITE, or DELETE statement is present
- SCAN statement is present
- The file is the target of a data lookup

If you are experiencing problems with a process in APPX 3.2 or later, and you know that process ran fine on a version of APPX prior to 3.2, try setting APPX_ACCESS_ALL_FILES=1 and see if the problem goes away.

Valid values: 1

Default value: not set

APPX_DISABLE_DYNAKEYS

Disables the ability of end users to create Dynamic Keys to data files if set to 1. Dynamic Keys were added in a 3.x release of APPX and sometimes sites have huge files that require a lot of processing and disk space to create a dynamic key. For this reason, a site might choose to not implement this feature on their production system, or might choose to restrict certain users from using it by setting this environment variable.

Valid values: 1

Default value: not set

APPX_DYNA_LOG

Logs Dynamic Key generation debug information. If you have generated a Dynamic Key and you get no records using the resulting key, or experience some other problem related to Dynamic Keys, set this variable to collect more information about what is happening during Dynamic Key generation. If this variable is not set, no debug information is collected.

Valid values: path to a write-enabled log file on your system

Default value: not set

APPX_NO_NEG_INT

Causes APPX to revert to pre-3.3 semantics when assigning a negative number to a signed binary. Assigning a negative number to a signed binary would cause an overflow in APPX releases prior to 3.3. If you actually do want the overflow to occur, you can return to the old behavior by setting APPX_NO_NEG_INT to 1.

Valid values: 1

Default value: not set

APPX_OLD_PD_MUL

Disables fixes to packed decimal multiplication that were made to APPX version 3.0. This variable probably should not be used; it is included here only for completeness.

Valid values: 1

Default value: not set

APPX_SCAN_PROC

Instructs APPX to use an alternate SCAN process. This variable was made obsolete by the use of 0SA ‘Hook’ records in APPX 4.0 and later.

Valid values: name of alternate SCAN process

Default value: new SCAN process, in 4.0 and later

APPX_VERIFY_IO

Turns off I/O verification if set to '0' (the digit zero, without quotes). If the environment variable is not defined at all, I/O verification is turned on for Windows, and turned off for Unix. You can turn this off if you are confident of your Windows network file access. If you are using Presentation Server on Windows you may want to turn this off, because there is a performance penalty associated with I/O verification which shows up under heavy user load as an increase in I/O counts and CPU utilization.

(Use of this variable may be required because Microsoft Windows network file sharing code has some bugs that have been known to corrupt data in APPX and other database file environments.)

Valid values: 0

Default value: not set

General Variables

APPXPATH

Denotes the default location of the APPX system administration files, applications and database files. You can override the location of the system administration files using the APPX_0SA_PATH variable. Additionally, you can override the location of any files by using FMS groups.

Valid values: path of the APPX data files directory on your system

Default value: *{appx engine location}*/data in Unix, *{appx engine location}*\data in Windows

APPX_0SA_PATH

Directs a PC network node to look at a server's disk drive for its APPX system administration files. One situation in which you may want to do this is when running several copies of APPX on networked PC's while sharing a single set of system administration settings like Users, Printers, Forms, etc.

To tell a PC network node to store its application design files on a server's disk drive but keep EMs on the local PC drive (for performance), define an FMS group as follows:

FMS Path: R:\APPX\data

FMS Controls: EM=C:\APPX\local_data

In this arrangement, FMS Path is the path to the APPX applications stored on the server's disk and the "EM=" value is a path to an existing directory on the local PC into which EMs can be cached. Then, assign the application design files to this FMS group for any application whose EMs you would like cached in this fashion.

Valid values: path to a server's disk drive in which the 0SA data directory is found, such as "R:\APPX\data"

Default value: APPXPATH directory

APPX_ALLOW_MISALIGNS

Converts ERROR messages to WARNING messages when the Data Dictionary is processed. In APPX releases prior to 4.1, numeric fields specified as BINARY in the field's Additional Attributes must reside on a byte boundary that is an even multiple of four. APPX generates an ERROR condition when processing the Data Dictionary if this condition is not met. Setting this variable to a non-blank value generates WARNING messages, which allows the Data Dictionary to be processed normally.

Valid values: 1

Default value: not set

APPX_DYNAKEY_PROMPT

Sets a threshold of records processed during Dynamic Key creation. When this threshold is reached, APPX displays a warning message and asks the user if they want to continue building the Dynamic Key, or if they want to cancel the Dynamic Key creation process. The default is 1000 records. To change the default, set this variable to the number of records at which you want the warning to appear. Note that this is an absolute value, not an interval; if you set this to 2000, you will not receive the warning every 2000 records. The warning and cancel opportunity will be displayed just **once**, after the 2000th record is processed.

Valid values: integers

Default value: 1000

APPX_END_PARAGRAPH

This environment variable tells APPX to use hex(0A) as an end-of-paragraph character within text fields instead of hex(B6). Databases such as Oracle expect to find hex(0A) within text fields and so does UniCode which is needed for entering and displaying Chinese, Japanese, and a few other languages.

This will also tell the server to notify the client on handshake to stop switching end of line characters when moving text data between the client and the server.

Export and Import also take notice of this flag and, if set, Export will switch the override character (0x0A in this case) with 0xB6 for text field data going into the portable file and Import will do the opposite conversion bringing the text data back in.

Valid values: 0x0A

Default value: 0xB6

APPX_KEYMAP

Sets the name of the default keymap. This is the same as using the "-m=" Command Line Option. See Appendix C of this manual for more information on Command Line Options.

Valid values: any valid keymap name for your APPX installation

Default value: not set

APPX_MSG

Centers the value of this variable on the bottom line of the screen. Setting this variable to `hostname` (Unix only) and/or $APPXPATH is useful. (Warning: if you want to use `hostname`, it must be set at the shell level in .profile or a similar place, not in appx.env. Only the shell can interpret `backticks`.)

Valid values: any text string, or `command` (where the output of command is the display text)

Default value: none

APPX_SHOW_PROGRESS

Activates a running status message during Query and related Output and Update processes. This message will keep you informed about the progress of the batch process. It shows the number of records processed, the number of records selected, etc. The message is updated every record until it reaches 10, then it is updated every 10 records until it reaches 100, then it is updated every 100 records, etc.

Valid values: 1

Default value: not set

APPX_STD_PGH_MASK

Sets the date mask for dates displayed in page headings. Setting APPX_STD_PGH_MASK to “06/05/0807 - 04:03 02.01” causes your page heading dates to look like “03/25/1999 - 00:18 13.82”.

Valid values: valid APPX date mask

Default value: system or application date mask, if set

APPX_ZERO_MALLOC

Fixes a bug in APPX version 2.x (sort of). This variable should no longer be set for sites running APPX 3.0 or later; the bug was fixed in version 3.0. Since there is a slight performance impact to the workaround, if you do not **need** this variable (i.e., you are not running a pre-3.0 version of APPX), don't set it. This is included only for completeness.

Valid values: 1 (set only if running a pre-3.0 version of APPX)

Default value: not set

Import/Export Variables

APPX_PORTABLE_DATE

Ignores "Bad portable date" or "Bad internal date" errors that may appear when importing SPEED II files containing dates. These errors can occur because SPEED II can store invalid dates in a date field via ILF code, and APPX rejects invalid dates during import.

Valid values: 1

Default value: not set

KEATERM Configuration Variables

APPX_KEATERM

Simulates GUI functionality if APPX_KEYMAP=keaterm. This variable determines whether or not a set of “smart buttons” is displayed by the terminal emulator. The use of the KEA terminal emulator to provide limited GUI functionality has been superseded by our true GUI clients. This variable is included only for completeness.

Valid values: Buttons, NoButtons

Default value: not set

Presentation Server Variables

(appxd and WinAppxD)

APPX_DISABLE_TCP_NODELAY

Disables the use of the TCP “NODELAY” option. The NODELAY option sends out characters without buffering multiple characters into a single packet. Buffering is what normally takes place. Disabling NODELAY may improve your APPX Presentation Client or APPX/Net performance. (It also may not. Results are dependent on your network configuration and what is happening across the network.)

Valid values: 1

Default value: not set

APPX_NET_FEATURE_MASK

Allows you to hard code a list of supported client/server features, in case you want to override the client or server's default list. For example, if both your client and server support the new Data Palette feature, but you do not want to make this feature available, you can disable it by eliminating the bit representing that feature from the feature mask on the client or server. This variable should be used only as directed by APPX Technical Support.

Valid values: hex numbers between 0x00000000 and 0xFFFFFFFF

Default value: APPX version-dependent

APPX_SERVER

Must be set to a valid filename of an APPX engine. For example, if APPX is installed in /usr/appx, set APPX_SERVER=/usr/appx/appx. On Unix systems, in most cases, this variable should be set in the start-appxd.sh shell script. On Windows systems, set it in the System environment variables dialog box. This variable must be set **before** appxd or WinAppxD is started. Changes made to it when appxd and WinAppxD are already running will be ignored until appxd or WinAppxD is killed and restarted.

Valid values: valid path to an APPX engine

Default value: not set

APPX_SRVR_TIMEOUT

Specifies the interval in minutes between server pings asking "is the other system alive?" in an attempt to detect dead connections. The default is 10. If you set it to a number below 5, APPX will reset it to 5 to avoid filling the network with constant "are you there?" queries.

Valid values: integers

Default value: 10

Printing and Job Submission Variables

APPX_BG_SCRIPT

Tells APPX to use a different script to submit batch jobs marked as "Separate Task? = Y". This is commonly used to interface APPX to batch job scheduling systems. APPX_BG_SCRIPT assumes a default starting point of APPXPATH. Any directories it specifies must therefore be a lower level under the "*{appx.engine location}*/data" or "*{appx engine location}*\data" directory.

Valid values: path to valid batch job submission script or .exe

Default value: not set

APPX_PREFIX

Specifies the drive prefix of the APPX installation on an NT Server for the UniQue Print spooler (under APPX for Windows). For more information, see www.appx.com/unique-nt.html or www.appx.com/unique-nt-pres.html.

Valid values: //C/, //D/, etc.

Default value: not set

APPX_PRT_FI_DIR

Sets the disk location where print files are created and stored on your system. **This variable must end with a forward slash (/) on Unix, and a backslash (\) on Windows.**

Valid values: path to a write-enabled directory

Default value: %APPXPATH%\print\{userID} (Windows) or /tmp/ (Unix)

APPX_PRT_FI_SIZE

Specifies how much space to pre-allocate for a print file, for VMS only. Since APPX is no longer supported on the VAX, this is included only for completeness.

Valid values: integers

Default value: not set

APPX_PRT_SCRIPT

Tells APPX to use a different script or .exe to process print jobs. In Unix, this variable is commonly set to "appx_print.UniQue.Unix" when users are running the UniQue print spooler. **The script or .exe must be in the APPXPATH directory.** Attempting to fully qualify the script's location with a full path (starting with / or \) will not work.

Valid values: valid print script or .exe within the APPXPATH directory

Default values: %APPXPATH%\winprint.exe (Windows), $APPXPATH/appx_print (Unix)

APPX_PRT_SCRIPT_HIDE

Invokes the APPX_PRT_SCRIPT without displaying a visible window on the screen (Windows only).

Valid values: 1

Default value: not set

APPX_SUPPRESS_UID_SWITCH

Tells APPX not to change print file ownership; all print files will remain owned by 'appx' (Unix only). Normally, when APPX creates a print file it changes the print file's ownership to the login ID of the user that generates the file. This provides a measure of security so that multiple users are unable to read each other's print files. In some situations you may want all print files owned by 'appx' rather than individual users.

Valid values: 1

Default value: not set

Scripting and Regression Testing Variables

APPX_SCRIPT_IN

"Plays back" all keystrokes recorded in it into the invocation of the APPX engine, as if the user had manually pressed these keys. It also substitutes any "#$word" sequences it finds in APPX_SCRIPT_IN with the current value of the environment variable denoted by "word". For example, #$USER becomes "michael" if the environment variable USER is set to "michael".

Valid values: path to an existing script file

Default value: not set

APPX_SCRIPT_OUT

Records all user keystrokes (and selected Processes executed in the program flow) into an output file. This file is quite readable and is editable for the use of APPX_SCRIPT_IN (described above). It is initialized upon entering APPX. It is not populated until APPX is successfully exited.

Valid values: path to an existing script-recording file

Default value: not set

APPX_SCRIPT_STEP

Causes APPX to pause before each APPX OPTION key is sent from APPX_SCRIPT_IN. While paused, you can press certain keys to do special processing:

- c = continues running the remainder of the script without stepping
- p = pauses and lets the user take over the keyboard until 'Ctrl-r' is pressed
- e = terminates the script playback and lets the user take control of the session

Any other key will step forward in the script until the next APPX OPTION key is encountered.

Valid values: 1

Default value: not set

APPX_SCRIPT_STOP

Specifies the behavior of APPX when running with APPX_SCRIPT_IN set, and performing regression testing. When set to “ALL”, it causes APPX to exit when a recorded screen differs from an actual screen generated during regression playback. When set to “STATUS”, it causes APPX to exit when a recorded screen’s status area differs from the status area of an actual screen generated during regression playback.

Valid values: ALL or STATUS

Default value: not set

APPX_TST_DIR

Points to a directory that can capture screen images. If this variable is set, APPX records every screen image it encounters into this location. If you are running with APPX_SCRIPT_OUT set, it stores the images in a subdirectory named “expect”. If you are running with APPX_SCRIPT_IN set, it stores the images in a subdirectory named “actual”. The screens themselves are saved in files with a name starting with “screen000”; the last three digits are incremented for each successive screen.

Valid values: any write-enabled directory that contains write-enabled subdirectories named “expect” and “actual”

Default value: not set

Unix-Based APPX Client Variables

These five variables are used when running the Unix-based APPX/Client. They exist because Unix is unable to pop up the login box, as Windows-based Clients can.

APPX_HOST

Specifies the APPX server hostname to connect to. This hostname must be running appxd (Unix) or WinAppxD (Windows). This variable is used in conjunction with the APPX_USER, APPX_PSWD and APPX_PORT variables.

Valid values: hostname running appxd or WinAppxD

Default value: not set

APPX_PORT

Specifies the port to connect to on APPX_HOST, to begin an APPX Presentation Server client session. WinAppxD (Windows) or appxd (Unix) must be running on this port on APPX_HOST for the connection to succeed. This variable is used in conjunction with the APPX_USER, APPX_PSWD and APPX_HOST variables.

Valid values: port number on APPX_HOST where appxd or WinAppxD is running

Default value: not set

APPX_PSWD

Specifies the password used to authenticate APPX_USER on the remote system APPX_HOST. This variable is used in conjunction with the APPX_USER, APPX_HOST and APPX_PORT variables.

Valid values: password for APPX_USER on the remote system APPX_HOST

Default value: not set

APPX_REMAP

Triggers keyboard mapping if set to 1. This environment variable is used in Unix only.

Valid values: 1

Default value: not set

APPX_USER

Specifies the user ID used to authenticate the user on the remote system APPX_HOST. This variable is used in conjunction with the APPX_PSWD, APPX_HOST and APPX_PORT variables.

Valid values: user ID on the remote system APPX_HOST

Default value: not set

Windows Variables

APPX_BIG_BUTTONS

Specifies that the "big button" GUI style should be used for the APPX Win32 client ("appx -c").

Valid values: 1

Default value: not set

APPX_NET_HASP

Specifies a network security HASP to login to. This variable was made obsolete by new HASP processing which is local, and does not use a HASP server, but is documented here for completeness.

Valid values: network security HASP server

Default value: not set

HASPTTY

Identifies the serial port that the HASP is attached to. To install a HASP on a Linux or SCO Unix system running on Intel (PC) hardware:

1. Get a 25-pin to 9-pin serial port adapter. (The HASP has 25 pins; PC serial ports have 9 pins.)
2. Install adapter and HASP onto PC serial port.
3. Set the environment variable (in autoexec.bat, or in $APPXPATH\appx.env):

HASPTTY=/dev/cua0

(If you install the HASP on the second port of a 2-port PC, the address will be /dev/cua1.)

1. Run APPX. No driver is needed. The serial number in the upper right hand box of the registration screen should match the HASP ID.

Valid values: /dev/cua0, /dev/cua1

Default value: not set

Y2K Option Variables

APPX_FAKE_TIME

Causes APPX to think the current date/time is something other than the current system clock's date and time. You can set this variable to "fake out" APPX for Y2K testing. The date and time specified in this variable will be used at the start of the APPX session. Time will tick forward from that point, so that an hour later, the time reported by APPX will be an hour later than the time it started with. This will not "fix" a set date and time for the entire session, just create a starting point for it. APPX_FAKE_TIME is supposed to set the time as well as the date, but multiple users have reported that while the time is reset, it is reset to some random value rather than the one requested. This variable should be set to a hard-coded date in APPX format. For example, to establish a date of 6/15/2001at 10:00am, set the variable this way: APPX_FAKE_TIME=20010615100000.

Valid values: date/time in ASCII format

Default value: not set

APPX_PIVOT_ITEMS

Establishes the year you want masked date items to pivot around. This variable is available as of APPX version 3.5.2 and applies only to masked dates on input screens. See www.appx.com/y2kdoc.html.

Valid values: 00 to 99

Default value: not set

APPX_PIVOT_YEAR

Establishes the year you want masked date items and ILF statements to pivot around. To pivot 2-digit years to 19__ or 20__ as appropriate, set this variable to the cutoff year. This variable applies to both ILF statements and masked dates on input screens. See www.appx.com/y2kdoc.html.

Valid values: 00 to 99

Default value: not set

DATABASE (RDBMS) INTERFACE VARIABLES

RDBMS Configuration Variables

RDBMS Configuration Variables

APPX_NO_END_CONSTRAINTS

Disables a performance optimization to use constraints on the upper range limit for record access. Do not set this variable unless requested by APPX Software, since the performance optimization is very useful.

Valid values: 1

Default value: not set

APPX_RDBMS_NDK

Superseded by the new RDBMS-related options that can be entered for FMS Groups and in File Specifications on the Database Management menu. These options are more flexible than environment variables and are the preferred choice. This information is included for completeness.

Determines how APPX creates RDBMS date key segments. Starting with release 3.5, you have a choice of how these date key segments are created. APPX_RDBMS_NDK controls the treatment of date key segments as follows:

Not Defined Date key fields are defined as ‘char’ and any NULL values are translated to or from blanks. This is a high performance option and also the default value.

APPX_RDBMS_NDK=NULL Date key fields are defined as 'date null'. This is a low performance option, and many of the queries produced by APPX will ignore the defined indices.

APPX_RDBMS_NDK=MANDATORY Date key fields are defined as 'date not null'. This is a high performance option, but no records with NULL values in a date key segment are stored.

APPX_RDBMS_NDK=some_value Date key fields are defined as 'date not null'. Instead of storing NULL values, 'some_value' is stored when writing to the RDBMS and translated into NULL when reading from the RDBMS. This value should be a date constant which is valid in the RDBMS. For example, 'some_value' could be defined as '00010101000000' to use midnight of January 1st, 0001 AD as your NULL date.

Valid values: NULL, MANDATORY, or a string to be used in place of NULL

Default value: not set

APPX_SQL_BUFFER

Specifies the size of the SQL command buffer (in bytes) when communicating with an RDBMS. This is the area that will hold the actual SQL commands that are sent to the database server (an INSERT, complete with the entire record contents to be INSERTED, for example). This variable should never be set lower than 2048. You may need to increase it beyond the default value if you have a table containing many large fields.

Valid values: integers

Default value: 16384

RDBMS - ISAM Configuration Variables

(Vision/CISAM)

APPX_PD_SIGNS

Tells APPX what byte pattern to use for the sign bits of packed decimal numbers. The first hex digit signifies the bit pattern for positive numbers, and the second the bit pattern for negative numbers. The default is "CD" which is how APPX is normally distributed. Vision uses a default of "FD" which can be overridden when the vision programs are compiled. The setting of this variable affects files stored as native APPX files. It also affects the APPX EMs, which store some packed decimal information. To use a setting of "FD", you must install a special set of APPX internal EMs, and if you have existing data in "CD" format, you must export the data, set APPX_PD_SIGNS=FD, then import the data. Note that if you are accessing Vision/CISAM files over a network using APPX/ODBC or APPX/Net, this variable must be set to the same value on both the client and the server systems.

Valid values: CD or FD

Default value: CD

RDBMS - ODBC Variables

(SQL Server, Access, etc.)

APPX_ODBC_AUTOCOMMIT

Enables the auto-commit flag for SQL Server. If you are accessing SQL Server data from APPX, you should set this variable to enable auto-commit. Superseded by the new RDBMS-related options that can be entered for FMS Groups and in File Specifications on the Database Management menu. These options are more flexible and are the preferred choice. This information is included for completeness.

Valid values: 1

Default value: not set

APPX_ODBC_ISOLATION

Tells APPX to request transaction isolation level 0 from the ODBC database being accessed. (Isolation levels may or may not be supported; this just requests it.) Setting this variable allows APPX to read data that has not yet been committed, which can avoid some potential deadlock situations. Do not set this variable unless requested by APPX Software.

Valid values: 1

Default value: not set

APPX_ODBC_LIB

Specifies the ODBC driver manager DLL location, for dynamic linking. The default should almost always be sufficient, but if not you can override the default by setting this variable. For example, if the ODBC driver manager DLL is not in the %WINDOWS_ROOT% directory, set this to the full path of its location.

Valid values: path to ODBC driver manager DLL

Default value: odbc32.dll

RDBMS - Oracle Variables

APPX_ORACLE_CONNECT

Specifies what user ID and password to use to login to Oracle. The database being requested can also be specified. The format of the string depends on the version of the server, the network library being used, the operating system, and other factors. Knowing how to login to Oracle from the command line is necessary before you can tell APPX how to do it.

This variable has been superseded by FMS control strings, but still exists within APPX. It is more flexible to use FMS control strings, and that is a better option. This is presented here for completeness.

Valid values: installation-dependent connect string

Default value: not set

APPX_ORACLE_FIRST_ROWS

"Hints" to Oracle that you want it to optimize its query processing so that the first few rows of a query are returned quickly. The alternative is to optimize the queries so that *all* rows are returned as quickly as possible. APPX generally displays a screen full of records and then stops, awaiting user interaction (rather than retrieving all records immediately). Therefore, setting this variable can improve Oracle's performance.

Valid values: 1

Default value: not set

APPX_ORACLE_LIB

Specifies the Oracle library location for dynamic linking. Usually the preset Oracle-based ORACLE or ORAHOME environment variables point to the home directory for the user 'oracle'. In some cases, however, you may want APPX to dynamically link against a set of libraries that are not in the usual place (for example, when testing a new version). In that case, set APPX_ORACLE_LIB to the alternate location, and it will override Oracle's setting.

Valid values: path to Oracle library

Default value: not set

APPX_ORACLE_PARAM_INSERTS

Specifies that parameterized INSERTs should be used when inserting data into Oracle tables. This variable has been superseded by the new RDBMS-related options you can enter for FMS Groups and in File Specifications on the Database Management menu. These options are more flexible than environment variables and are the preferred choice. This information is presented for completeness.

Valid values: 1

Default value: not set

APPX_ORIO_LIBS

Specifies the names of the Oracle libraries that APPX references when dynamically linking in Oracle functionality. The default list should be sufficient, but may need to be altered for different versions of Oracle, or for the same version on different operating systems. If APPX complains about a symbol not found, after you have installed a new version of Oracle, that is a symptom of the need to change the list of libraries.

Valid values: list of Oracle libraries containing functions used by the APPX/ORACLE interface

Default value: libnet.so,libclntsh.so

RDBMS - Sybase Variables

APPX_SYBASE_DATABASE

Specifies a default Sybase database name. This variable is superseded by the new RDBMS-related options you can enter for FMS Groups and in File Specifications on the Database Management menu. These options are more flexible than environment variables and are the preferred choice. This information is presented for completeness.

Valid values: existing Sybase database name

Default value: not set

APPX_SYBASE_LEVEL0

Enables “Level 0” table locking in Sybase. This is still a development feature and should not be enabled unless requested by APPX Software.

Valid values: 1

Default value: not set

APPX_SYBASE_LIB

Specifies the Sybase library location for dynamic linking. Normally there is a preset Sybase-based environment variable that points to the home directory for the user ‘sybase’. In some cases you may want APPX to dynamically link to a set of libraries that is not in the usual location (for example, when testing a new version). In that case, set APPX_SYBASE_LIB to that alternate location, and that will override Sybase’s setting.

Valid values: path to Sybase library

Default value: not set

APPX_SYBASE_PACK_SIZE

Determines the packet size for Sybase database server communications. Setting this variable to 4096 may improve communications efficiency.

Valid values: integers

Default value: not set

APPX_SYBASE_SERVER

Specifies the default Sybase server name. This variable has been superseded by FMS group settings, and is included here only for completeness.

Valid values: valid Sybase server name

Default value: not set

RDBMS - Vision Variables

APPX_ISKEYSEG

Overrides the normal APPX behavior of trying to consolidate multiple key segments. Instead, if this variable is set, APPX forms key segments in the way Vision does through ACUCOBOL. If Create Files in Database Management fails by returning to Database Management without creating the files selected, setting this variable may correct the error. Likewise, if you receive the APPX error FMFI.C “can't empty pfb list”, setting this variable may correct the error.

Valid values: 1

Default value: not set

APPX_VISION_LOCKS_PER_FILE

Specifies the maximum number of locks per file, per process. You may need to increase this to 20 or greater for applications containing complex processes that use CISAM files. This variable is available beginning with APPX 3.2.4.

Valid values: integers

Default value: not set

APPX_VISION_MAXFILES

Specifies the maximum number of open files in the Vision file system. You may need to increase this to 255 or greater for applications containing complex processes which use CISAM files. This variable is available beginning with APPX 3.2.4.

Valid values: integers

Default value: not set

APPX_VISION_MAXLOCKS

Specifies the maximum number of record locks per process. You may need to increase this to 40 or greater for applications containing complex processes which use CISAM files. This variable is available beginning with APPX 3.2.4.

Valid values: integers

Default value: not set

DEBUGGING VARIABLES

Debugging - APPX/Net Variables

Debugging - APPX/Net Variables

APPX_NET_HEADERS

Enables APPX client/server network packet headers for debugging.

Valid values: 1

Default value: not set

APPX_NETIO_CODE

Specifies the debug level for APPX_NETIO_LOG. If you set APPX_NETIO_LOG, you should almost always set this to 0xFFFFFFFF.

Valid values: hex numbers between 0x00000000 and 0xFFFFFFFF

Default value: 0x00000000

APPX_NETIO_LOG

Specifies the log file name into which APPX/Net debugging information will be written, on the client side of an APPX/Net connection. If you experience problems with APPX/Net, set this variable and its companion debug level variable, APPX_NETIO_CODE, and client-side debugging information will be written to this file.

This is the higher level of a two-level protocol; the lower level is specified by APPX_NET_LOG. Both should be enabled when debugging APPX/Net or APPX/ODBC. It may also be helpful to set APPX_NETIO_SERVER and APPX_NET_LOG_MASK. See Debugging APPX/Net or APPX/ODBC (Remote) remarks for additional information.

Valid values: path to a write-enabled log file on the client

Default value: not set

APPX_NETIO_SERVER

Specifies the log file name into which APPX/Net debugging information will be written, on the server side of an APPX/Net connection. If you experience problems with APPX/Net, set this variable and server-side debugging information will be written to this file.

Valid values: path to a write-enabled log file on the server

Default value: not set

Debugging - APPX/ODBC Variables

APPX_ODBC_CODE

Specifies the debug level for APPX_ODBC_LOG. The default value means “log everything”.

Valid values: hex numbers between 0x00000000 and 0xFFFFFFFF

Default value: 0xFFFFFFFF

APPX_ODBC_LOG

Specifies the log file name into which the APPX/ODBC driver will write debugging information. If you experience problems with APPX/ODBC, which is the driver that accesses APPX files from PC client applications like MS Access, this variable will contain a trace of APPX/ODBC’s activities. If you want the log to contain less than all tracings, you must also set the corresponding debug level variable, APPX_ODBC_CODE.

> Note that both APPX_NETIO_LOG and APPX_NET_LOG should be enabled when debugging APPX/ODBC. See Debugging APPX/Net or APPX/ODBC (Remote) remarks for additional information.

Valid values: path to a write-enabled log file

Default value: not set

Debugging - APPX CGI Interface Variables

APPX_CGIDATA_LOG

Specifies the log file name into which CGI data parsing debug information is written. If you are running APPX using the CGI interface, and want to see exactly what variables are being received by the APPX process during the CGI data import, set this variable to the name of a valid log file. The CGI data parsing routine will record the variables and values it finds on its input stream in the designated file.

Valid values: path to a write-enabled log file

Default value: not set

Debugging - CISAM Variables

APPX_CISAM_DEBUG

Activates CISAM debug options. If you experience problems with CISAM file access, this variable will record detailed error messages from CISAM to aid in problem identification.

Valid values: path to a write-enabled log file

Default value: not set

Debugging - License Server Variables

APPX_LS_DBG

Specifies the debug level for license server issues. If you experience license server problems, set this to 0xFFFFFFFF. Debug information for the “appx -l=” license server process will be written to the file named after the “-l=”, and debug information for clients attempting to obtain session licenses will appear in APPX_LS_LOG, if it is set. You must kill and restart the license server process in order for this variable setting to affect it. Setting this variable after the license server is running will not cause it to log anything. **Warning:** license server logging can issue many messages, so pay attention to the size of your log file when you have this variable set. You can fill up your disk more quickly than you think.

Valid values: hex numbers between 0x00000000 and 0xFFFFFFFF

Default value: 0x00000000

APPX_LS_DELAY

Specifies the number of seconds to wait after starting the license server before an APPX session validates its license information. If you receive a “no license server detected” error message upon starting APPX, set this variable to a larger number than the default value. This will avoid that error message, at a slight performance cost.

Valid values: integers

Default value: 5 (Unix) or 8 (Windows)

APPX_LS_GHOSTS

Clears out “ghost” sessions from the 0SA USAGE file. “Ghost” sessions are still listed in the USAGE file, but their actual APPX processes no longer exist. This can happen any time an APPX session is abnormally ended. Aggressively pursuing the removal of these on busy systems results in excessive contention for USAGE locks, so this variable is now disabled as a default.

If a site has no problems with USAGE locks, and would like to re-enable the algorithm that clears “ghosts”, set this variable to 1 for that site. Their display of license users will then more closely match reality. Having this variable disabled can result in the buildup of a large number of entries in the USAGE file, which can reduce system performance.

It is necessary to restart your license server for this variable to take effect.

Valid values: 1

Default value: not set

APPX_LS_INTERVAL

Specifies the number of minutes between checks of USAGE file information. Sometimes performance on busy systems can be improved by setting this to a larger number than the default value. Do not set this variable to a number greater than 10 minutes.

Valid values: integers

Default value: 2

APPX_LS_LOG

Specifies where license server client debugging information should be written. In older releases of APPX (early 3.x) this variable was used to specify the server log file location as well. However, all license server debugging information is now written into the log file named after “-l=” on the license server command line. See the variable APPX_LS_DBG for additional information.

Valid values: path to a write-enabled log file

Default value: not set

Debugging - Memory Variables

APPX_CHECK_MEM

Causes APPX to periodically pause and verify its allocated memory, in an attempt to narrow down the source of memory corruption. This process is very CPU-intensive, and will noticeably slow your APPX session. Therefore, do not use this variable unless specifically requested by APPX Software.

Valid values: 1

Default value: not set

APPX_RUN_MEM

Overrides the amount of memory APPX will use for an in-memory query. Above this value, APPX will switch to a disk-based sort, which is less efficient. You may want to increase the default value if queries that run well with a small number of records slow noticeably when the number of records increases.

Valid values: maximum number of bytes to use for an in-memory query sort

Default value: 256K on Unix, 64K on Windows

APPX_TRAILER_COUNT

Tells APPX to append additional trailer bytes to each memory allocation. These bytes are filled with known values that are later checked for corruption when the memory is given back to the system. They can be used to find out if APPX is writing beyond the end of an allocated chunk of memory, or if memory is being corrupted for some other reason. If you set APPX_TRAILER_COUNT and memory corruption is detected, you will receive an "mm.c.876 mm_free() – corrupted" error message as soon as the corruption is detected. This variable may introduce a small amount of additional memory overhead.

It is also useful to set this variable if you are receiving other "mm.c" error messages, because it may detect the source of memory corruption before the other error message occurs, and thus provide better debugging hints. This variable can also provide information for ".stk" stack dumps, if they are enabled on your system.

Valid values: integers

Default value: 0

Debugging - General Variables

APPX_DB_TRAPS

Enables the use of 'Ctrl-C' as a dynamic trap when you are running processes outside of Application Design. Set this variable if you wish to allow a non-designer to use 'Ctrl-C', or if you wish to allow a designer to use 'Ctrl-C' when running processes outside of Application Design. This variable also enables TRAP statements when outside of Application Design.

Valid values: 1

Default value: not set

APPX_DBG_VFY

Specifies the log file name into which APPX writes the results of Executable Module rebuilding checks. If an EM appears to be rebuilding too frequently (for example, every time the process is invoked), set this variable to record the reasons the EM is being rebuilt. That may indicate a solution to the problem.

Valid values: path to write-enabled log file

Default value: not set

APPX_FD_CACHE_CODE

Determines how much is written to the file specified by APPX_FD_CACHE_LOG. A value of 1 will record file opens. A value of 3 will provide a history of all record locks that APPX acquires and frees.

Valid values: 1, 3

Default value: not set

APPX_FD_CACHE_LOG

Specifies the log file name into which all file opens and their success or failure status will be written. If APPX is behaving unpredictably and the reason is not obvious, this log may be helpful. You may have a permissions problem, or perhaps a missing structure file, or a corrupted data file. This log file can often indicate an area to investigate. **Warning:** this log file can become fairly large. The most efficient way to read it is to bring it up in an editor, go to the bottom, and read up, looking for failures. If you see the phrase “giving up (number)”, it means that APPX could not open that file. The reason for the failure is indicated by the number in parentheses. You must also set the APPX_FD_CACHE_CODE variable, to set the debug level for this log.

This variable may actually log when files are closed (not when they are opened), and some I/O statistics associated with each file.

Valid values: path to a write-enabled log file

Default value: not set

APPX_FMT_IO_LOG

Causes the IO log to be formatted for readability.

Valid values: 1

Default value: not set

APPX_IO_LOG

Specifies the log file name into which APPX will write all file I/O, keystrokes and Process ID’s. This variable is used only for APPXIO files.

Valid values: path to a write-enabled log file

Default value: not set

APPX_UNIXIO_PROCS

Causes APPX to also log process names as they are executed. This variable is used in conjunction with APPX_UNIXIO_STATS; if APPX_UNIXIO_STATS is not set this variable does nothing. This variable is available only on Unix systems.

Valid values: 1

Default value: not set

APPX_UNIXIO_STATS

Specifies the log file name into which APPX will write every APPXIO file open, close and I/O. APPX opens and immediately closes some files during startup. If you set this variable, and try to run APPX, and the log file is empty, there must be some generic error occurring. If you get some entries in the log, then the problem is with one or more, but not all, files. This may help you catch permission errors. Note that the statistics for a file are written when the file is closed, so if a file is closed and reopened, it will appear more than once in the log. This variable is available only on Unix systems.

For each I/O type (OPEN, READ, READNEXT, START, WRITE, REWRITE), the format of the report it produces is:

TOTAL-Time (seconds.microseconds), LONGEST-Time (seconds.microseconds), COUNT

Valid values: path to a write-enabled log file

Default value: not set

track_locks

Activates the APPXIO feature to keep track of lock holders on a file-by-file or FMS Group basis. This is not an environment variable, but is placed here because this is a likely place to look for this capability.

If you include the text ‘track_locks’ in an FMS control string, APPX records extra information for this file whenever a record is locked. There is a performance penalty when you use this option (two extra I/O operations for each successful read with hold) so you may want to apply this option selectively.

The three-character User ID will be displayed whenever APPX waits for more than 10 seconds for a lock. Note that APPX will not wait for the PCF in an INPUT process, so you will not see the User ID in that situation unless you have ILF statements within the process of READ, BEG READ, or READNEXT that refer to the PCF.

On Unix systems, you will still see a process ID if you don't use ‘track_locks’. There is no performance penalty for this.

Debugging - Presentation Server Variables

APPX_NET_LOG

Specifies the log file name into which APPX will write debugging information. If you experience problems with the Windows or GUI Client, APPX/Net, or APPX/ODBC connecting to your appxd or WinAppxD APPX/Servers, set this variable and APPX_NET_LOG_MASK on both your server and client systems to generate debug logs.

It is most useful to do this on the server, since usually Presentation Client/Server startup issues involve user authentication or security, especially on NT. However, it is useful to do it on the client, too.

On NT especially, look for a line of the form: "LogonUser() failed with {number}". If you see this line, please let APPX Technical Support know which error number you are receiving.

This is the lower level of a two-level protocol; the higher level is specified by APPX_NETIO_LOG. Both should be enabled when debugging APPX/Net or APPX/ODBC. See Debugging APPX/Net or APPX/ODBC (Remote) remarks below for more information.

Valid values: path to a write-enabled log file

Default value: not set

APPX_NET_LOG_MASK

Specifies the debug level for APPX_NET_LOG. When debugging, this variable should almost always be set to 0xFFFFFFFF.

Valid values: hex numbers between 0x00000000 and 0xFFFFFFFF

Default value: 0x00000000

Debugging APPX/Net or APPX/ODBC (Remote)

When debugging APPX/Net or APPX/ODBC to remote APPX data problems, there are two levels of APPX/Net protocol that can be logged: APPX_NET_LOG (the lower level) and APPX_NETIO_LOG (the upper level). Protocol is the mechanism by which the client and server talk to each other.

The lower level (NET LOG) is common to both APPX/Net and the APPX Presentation Server.

Lower level protocol on the client is monitored by using the environment variables APPX_NET_LOG and APPX_NET_LOG_MASK. For example:

APPX_NET_LOG=C:\net.log

APPX_NET_LOG_MASK=0xFFFFFFFF

Lower level protocol on the server side is monitored by using the environment variables APPX_NETIO_SERVER and APPX_NET_LOG_MASK. For example:

APPX_NETIO_SERVER=/tmp/netio.server.log

APPX_NET_LOG_MASK=0xFFFFFFFF

Upper level protocol on the client side is not being logged as of 4.0.a.

Upper level protocol on the server side is monitored by using the environment variables APPX_NETIO_LOG and APPX_NETIO_CODE. For example:

APPX_NETIO_LOG=/tmp/netio.log

APPX_NETIO_CODE=0xFFFFFFFF

When running APPX/ODBC to remote APPX data, you also have APPX/ODBC logging on the client side:

APPX_ODBC_LOG=C:\appxodbc.log

APPX_ODBC_CODE=0xFFFFFFFF

Debugging - RDBMS Access Variables

APPX_DBG_CODE

Sets the debug level for APPX_SQL_CMD, described below. When it is needed, this variable is normally set to 0xFFFFFFFF.

Valid values: hex numbers between 0x00000000 and 0xFFFFFFFF

Default value: 0x00000000

APPX_SQL_CMD

Specifies an SQL Log into which APPX will write all SQL traffic from APPX and the RDBMS and back. The log records all commands APPX sends to the RDBMS, and all data the RDBMS sends back to APPX. This is a useful tool for diagnosing RDBMS back-end problems.

RDBMS's supported include Oracle, Sybase, SQL Server and MS-Access. The last two options use the APPX/ODBC interface.

Alternatively, create an ODBC log from Windows: Bring up the ODBC Administrator program used originally to set up the data source. In the "Tracing" tab, click on "Start Tracing" to start and "Stop Tracing" to stop. A trace must be started before your APPX session ever accesses the data file. If APPX tries to access the data and fails, you must completely exit APPX, and go back in, to get a trace.

Valid values: path to a write-enabled log file

Default value: not set

Debugging - Restructure Variables

APPX_TRAP_RESTRUCT

Inserts TRAP statements in Restructure event points for debugging purposes.

Valid values: 1

Default value: not set

APPX_VFY_RESTRUCT

Forces a restructure to occur when requested, even if APPX thinks that a restructure is not needed for a particular file. Do not set this variable unless instructed to do so by APPX Software. Setting it removes the “Restructure not required” message normally received during restructure. Any file you select to be restructured will be restructured. This can potentially waste a lot of time recreating files that are already synchronized with their Data Dictionary definitions.

Valid values: 1

Default value: not set

Debugging - Security HASP

APPX_SS_CODE

Specifies the debug level for APPX_SS_LOG. When this variable is needed, it is almost always set to 0xFFFFFFFF.

Valid values: hex numbers between 0x00000000 and 0xFFFFFFFF

Default value: 0x00000000

APPX_SS_LOG

Specifies the log file name into which security HASP initialization and verification debug information is written. When APPX tells you it cannot find your HASP, or if you receive an invalid serial number message when trying to run APPX on a system using a HASP, use this log to gather more information about why it is failing. Among other things, it provides a record of what HASP serial number APPX sees on your system. If you set this, also set “APPX_SS_CODE=0xFFFFFFFF” so that APPX will log the maximum amount of debug information.

Sample Output:

Calling for INIT_CRYPT_SYS...returned 32

Calling for GET_HASP_ID...returned 32 (id=7)

A return code of '32' means the environment variable HASPTTY was not set.

A return code of '129' means the OpenHASP key was not found.

Valid values: path to a write-enabled log file

Default value: not set

Debugging - Language Translation Variables

APPX_XLATE_LOG

Specifies the log file name into which APPX will write language translation debugging information.

Valid values: path to a write-enabled log file

Default value: not set

Debugging - Unix

APPX_SIGNAL_LOG

Specifies the log file name into which APPX will write error signals like bus errors and segmentation faults.

Valid values: path to a write-enabled log file

Default value: not set

APPX_SYS_ERR_LOG

Specifies the log file name into which APPX will write all system errors.

Valid values: path to a write-enabled log file

Default value: not set

Debugging - Vision Variables

VISION_VLOG

Activates Vision debug logging into the fixed file name "v.log". This log file will be written into the APPXPATH directory.

Valid values: 1

Default value: not set

Debugging - Windows Variables

APPX_NO_STACK_DUMPS

Disables stack dumps on Windows systems. **CAUTION:** Do not set this variable unless instructed to do so by APPX Technical Support, as stack dumps (the appx.stk file) are normally quite useful in diagnosing APPX software issues.

Valid values: 1

Default value: not set

APPX_PRELOAD_MAX

Avoids a "panic – memory allocation error" under certain circumstances. If you use APPX in a shared network disk configuration, with PC clients having APPXPATH set to a server's disk drive, you may need to set this variable. When APPX pre-loads an Executable Module, it first looks to see how big the EM is. When it does this across a network using Windows network file sharing, sometimes an incorrect number is returned. This is due to a bug in Windows. Since the number returned is usually quite large, APPX can try to allocate as much as 3 or 4 gigabytes of memory for that EM. This causes the above-mentioned memory allocation error. Setting this variable disables the pre-load, and therefore avoids the error.

Valid values: 1

Default value: not set

Debugging - winprint Variables

WINPRINT_DEBUG

Specifies a log file name into which APPX will write Windows printing debug information. If you experience problems with 'winprint.exe', this variable will direct debug information to the specified file (in versions of winprint.exe dated 12/95 or later).

Valid values: path to a write-enabled log file

Default value: not set

SETTING ENVIRONMENT VARIABLES

Global OS Environment Variables

Global OS Environment Variables

The “classic” and most general place to set environment variables is at the operating system level. In Unix, they are typically set in */etc/profile*. In Windows, use the NT system environment settings or shared autoexec.bat files.

Benefits

- Familiar to APPX veterans
- Convenient if you have a mixture of Telnet and APPX/Client users
- Easy to verify variable values with “set |more” at a command prompt

Disadvantages

- Variables tend to end up scattered around
- Using a mixture of operating system and appx.env variables leaves you with variable settings in any of many different places, and this can be hard to debug
- For non-APPX/Client users, the user must log out and log back in again before changes in variable values can take effect
- For APPX/Client users, the WinAppxD System Service must be stopped and restarted, or the NT server must be rebooted, before changes in variable values can take effect

Per-user OS Environment Variables

To modify the behavior of APPX at runtime for a specific user or users, add the environment variable settings in Unix's ~/.profile, or the Windows non-shared autoexec.bat file.

Setting environment variables in this manner:

- Is not applicable for APPX/Client
- Is for variables set on a per-user basis
- Is only useful in “shared disk” configurations on Windows, and for Unix Telnet access to APPX.
- Overrides any operating system settings
- Also overrides any global appx.env settings

Startup Script or Batch Files

The next technique for setting APPX-related environment variables is to do so directly in the script or batch file invoked by system users to start APPX, or invoked by the system to start appxd.

Benefits

- If the site already maintains a script or batch file to start APPX, this may be convenient
- Like appx.env, this does not require reboots or system logout/logins for changes to take effect, just exiting and re-entering APPX
- Already present in the start-appxd.sh file for client/server installations
- Only affects APPX/Client users if placed in the appxd start up script (Unix only)
- When placed in the appxd startup file, all variables can be placed in one file

Disadvantages

- If you do not already have an APPX startup script or batch file for some other site-specific purpose, you probably do not want to add one; it is another file to maintain and remember the purpose of
- Does not offer anything beyond appx.env's functionality, if you are not already doing other things in an APPX startup script that need scripting or batch file programming capabilities
- If someone ever runs APPX by invoking APPX.EXE directly, the script's variable settings will not be executed
- Only affects non-APPX/Client users
- When placed in the start-appxd.sh file, site administrators should be careful to back up this file prior to an APPX upgrade in case a new version includes an updated template start-appxd.sh file

appx.env Environment Variables

Global appx.env Environment Variables

The recommended method for setting environment variables is in the global *appx.env* file, which is in the APPXPATH directory (APPXPATH/appx.env).

Benefits

- Convenient if you have a mix of Telnet and APPX/Client users
- All variables are in a single file

Per-user appx.env Environment Variables

Per-user appx.env is not available for APPX/Client users and it is only useful in “shared disk” configurations on Windows, and for Unix Telnet access to APPX. The variables should be placed in either the ~/appx.env or C:\appx.env files.

Benefits

- Allows you to set variables on a per-user basis

Disadvantages

- Overrides any operating system settings; also overrides any global appx.env settings
- Not applicable for APPX/Client

Cautions in Setting Environment Variables

Observe these cautions when setting APPX environment variables:

- APPX_PRT_FI_DIR must have a trailing slash.

This variable determines where APPX creates report files. Examples in Unix and Windows:

$ export APPX_PRT_FI_DIR=/tmp/pat/

C:\> set APPX_PRT_FI_DIR=\\server\printfiles\pat\

- Make sure path names are valid.

For any variable requiring a file or directory pathname, make sure that all directories in the path exist. On Unix (or when editing appx_print for UniQue), make sure that the case you use in the variable setting matches the actual case of the filename. A mistake here can cause log files to not appear, or APPX to behave incorrectly.

- Setting a variable in .profile for an APPX/Client user will have no effect.

While a variable set in a Unix .profile will affect a user’s APPX session if they Telnet in to the APPX machine, it will not affect a APPX/Client login. Since we directly start up APPX from APPX/Client, rather than going through a Command Shell as with Telnet, no shell initialization files are processed.

- Clicking "Apply" after a change will not update a running copy of WinAppxD.

WinAppxD runs as a system service. As such, the only way to get WinAppxD (and subsequent APPX/Client sessions started from it) to recognize changes in the system environment is to reboot the server.