

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page vii

First Edition (December 1995)

This edition applies to the licensed program Application Dictionary Services/400 (Feature 2212), Version 3 Release 6 Modification 0, a feature of Application Development ToolSet for OS/400 (Program 5716-PW1), and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the proper edition for the level of the product.

Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change or addition.

Order publications through your IBM representative or the IBM branch office serving your locality. If you live in the United States, Puerto Rico, or Guam, you can order publications through the IBM Software Manufacturing Solutions at 800+879-2755. Publications are not stocked at the address given below.

A form for reader comments is provided at the back of this publication. If the form has been removed, you can mail your comments to:

IBM Canada Ltd. Laboratory
Information Development
2G/345/1150/TOR
1150 Eglinton Avenue East
North York, Ontario, Canada. M3C 1H7

You can also send your comments by facsimile (attention: RCF Coordinator), or you can send your comments electronically to IBM. See "Communicating Your Comments to IBM" for a description of the methods. This page immediately precedes the Readers' Comment Form at the back of this publication.

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you or restricting your use of it.

© **Copyright International Business Machines Corporation 1988, 1995. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks	vii
About This Book	ix
Who Should Use This Book	ix
What You Should Know	ix
Conventions Used in This Book	x
Chapter 1. Introduction	1
What is Integrated Language Environment?	1
Description of the ILE Program Objects	2
ILE Programs	2
Service Programs	3
Modules	3
What Is the Difference Between a Module and a Procedure?	3
The Application Dictionary Services/400 User Environments	4
What is an Application Dictionary?	4
What Is Automatic Refreshing?	4
Types of Information Stored in an Application Dictionary	5
Static Information	5
Derived Information	5
How Application Dictionary Services/400 Maintains Information in a Dictionary	6
For Deleted Objects or Parts	6
For Created Objects or Parts	6
Overview of the Application Dictionary Services/400 Functions	6
Supported Application Development Manager/400 Functions	7
Chapter 2. Getting Started	9
Installing Application Dictionary Services/400 and Transferring Dictionaries	9
Creating Your First Dictionary	10
Starting the AppDict Services/400 Tool	10
Using the STRADS Command	10
Using the STRADS Command in an Application Development Manager/400 Session	12
Starting At the Application Dictionary Services/400 Menu	14
Using Application Dictionary Services/400 As a Stand-Alone Tool	14
Using Application Dictionary Services/400 in an Application Development Manager/400 Session	14
Using the WRKAPPDCT Command	14
Refreshing Your Screen	16
Getting Help	17
Chapter 3. Making the Most of Your Application Dictionary Services/400 Working Environment	19
Changing Function Defaults	19
Creating Your Own Options	20
Running Multiple Sessions	24
Working with Subsystems	25
Chapter 4. Working with Dictionaries	27

Statuses of Dictionaries	27
Creating Dictionaries	28
Creating Dictionaries Directly in Application Dictionary Services/400	28
Creating Dictionaries in an Application Development Manager/400 Session	30
Creating Multiple Dictionaries	31
Adding or Removing the Documentation of Libraries and Groups	32
Documenting Additional Libraries or Groups in a Dictionary	32
Documenting Additional Libraries	32
Documenting Additional Groups	33
Removing the Documentation of Libraries or Groups from a Dictionary	34
Removing Documentation of Libraries	34
Removing Documentation of Groups	34
Reorganizing a Dictionary	35
Deleting a Dictionary	36
Finding Documented Libraries and Documenting Dictionaries	36
Supporting Multiple Projects with One Dictionary	37
Chapter 5. Creating Program Objects	39
How Does the AS/400 System Create Programs, Service Programs, and Modules?	39
Accessing Data Items and Procedures Across ILE Program Objects	40
Creating Programs, Service Programs, and Modules in Application Dictionary Services/400	40
Creating Files	46
Types of Files You Can Create	46
Creating a Field Reference File From a Template	47
Creating a Physical File For a New Field Reference File	47
Defining File-Level Keywords	49
Defining Record-Level Keywords	50
Defining Fields	50
Identifying Key Fields	53
Saving the Data Description Specifications Source	55
Creating a Logical File	56
Specifying the Types of Access for Files	57
Creating SQL Files	57
Creating an SQL Table File Based on a Field Reference File	57
Creating an SQL Index or View File Based on an SQL Table File	59
Creating an SQL Index or View File Based on a Physical File	61
Creating Other Types of Objects	61
Compiling and Building Program Objects	64
Compiling Objects	64
Building Objects	64
Chapter 6. Marking Field Reference Files	65
Marking New Field Reference Files	65
Unmarking Previously Marked Field Reference Files	68
Chapter 7. Managing the Impact of Changes to Fields	69
Determining the Field Reference Hierarchy	69
Determining Which Files Are Affected By a Change	75
Example	77
Determining Which Programs, Service Programs, and Modules Are Affected	78
Changing a Field	80
Recreating Impacted Objects	81

How the Recreate Function Works	81
Which Default Options Does AppDict Services/400 Use	81
Order in Which AppDict Services/400 Recreates Parts	82
Results of a Failure in the Recreation Process	82
Recreating Directly in AppDict Services/400	82
When Changes Have Been Made to Fields or Files	83
When Changes Have Been Made to ILE Modules or Service Programs	85
Recreating from Within an Application Development Manager/400 Session	85
Chapter 8. Finding Objects and Their Relationships to Other Objects	87
Finding Strings and Scanning ILE Program Objects	87
How to Find Strings and Scan RPG Source	87
Finding Strings in Source Members	88
Finding a Field Name in an RPG Program	89
Finding Programs, Service Programs, and Modules	90
Determining the Program Calling Structure	90
How is the Calling Structure Defined in the Integrated Language Environment	90
Determining the Calling Structure for OPM Programs and Procedures	91
Determining the Referencing Structure for ILE Programs and Objects	97
Finding Programs Referencing a Data Area	100
Finding Programs, Service Programs, or Modules That Reference a File	102
Finding Files	104
Finding Files Related to a Physical File	104
Finding Files Related to a Logical File	108
Finding Files Used By a Program, Service Program, or Module	109
Finding Information About Record Formats Defined in Files	111
Finding Data Areas Used By Programs, Service Programs, and Modules	114
Finding Objects By the Last Changed Source Date	116
Chapter 9. Maintaining the Application Dictionary Services/400 Tool	119
Ensuring Your Dictionaries Are Automatically Refreshing	119
More Technical Information About Automatic Refreshing	119
Determining Whether or Not Automatic Refreshing Is Working	120
First Method	120
Second Method	120
Rebuilding a Dictionary	121
Backing Up the Application Dictionary Services/400 System	122
Appendix A. Control Language Commands in Application Dictionary Services/400	125
STRADS (Start AppDict Services/400) Command	125
Purpose	125
Required Parameters	125
Optional Parameters	126
Example	126
WRKAPPDCT (Work with Dictionaries) Command	127
Purpose	127
Optional Parameters	127
Example	127
Appendix B. System/36 and System/38 Restrictions	129
Bibliography	131

Index 133

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594, U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independent created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Canada Ltd., Department 071, 1150 Eglinton Avenue East, North York, Ontario M3C 1H7, Canada. Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

400	Operating System/400
Application System/400	OS/400
AS/400	PROFS
C/400	RPG
COBOL/400	RPG/400
DB2/400	SQL
IBM	System/36
Integrated Language Environment	System/38
IBMLink	

About This Book

The Application Dictionary Services/400 (AppDict Services/400), a feature of the IBM Application Development ToolSet/400 (ADTS/400) product, helps you design and maintain both Original Program Model (OPM) and Integrated Language Environment (ILE) applications on the Application System/400 (AS/400) system.

This book explains how to use only the most commonly used functions by leading you through detailed examples. For additional information about the options, function keys, commands, and messages, refer to the online help provided with this feature.

Who Should Use This Book

This book is for application programmers or analysts who are:

- Doing impact analysis for host-based AS/400 applications
- Keeping track of their application objects (and the relationships that these application objects have to other application objects) as they develop them in an IBM AS/400 environment.

What You Should Know

It is assumed that you already know how to use your workstation, how to program in the AS/400 environment, and have a general knowledge of the AS/400 system.

It is also assumed that you are familiar with the Application Development Manager/400 feature, if you will be using the Application Dictionary Services/400 feature from within an Application Development Manager/400 session. Application Development Manager/400 is also a separately orderable feature of the Application Development ToolSet/400 product. For more information about Application Development Manager/400, refer to the *ADTS/400: Application Development Manager/400 User's Guide*, SC09-2133.

For information about other AS/400 publications, see either of the following:

- The *Publications Reference* book, SC41-4003, in the AS/400 Softcopy Library.
- The *AS/400 Information Directory*, a unique, multimedia interface to a searchable database containing descriptions of titles available to IBM or from selected other publishers. The *AS/400 Information Directory* is shipped with your system at no extra charge.

Refer to the "Bibliography" on page 131 for other books you may need for reference.

Conventions Used in This Book

Convention	Meaning
ALL CAPS	Commands, keywords, and names ¹
Bold type	New terms ²
<i>Italics</i>	<ul style="list-style-type: none">• Titles of manuals• Prompts on displays
Monospace type	<ul style="list-style-type: none">• Information that you type on a display¹• Messages on displays

Notes:

1. Although this manual always shows commands and entries in uppercase letters, you can use uppercase, lowercase, or mixed-case letters.
 2. This book does not contain a glossary; terms are defined when they first appear.
-

Chapter 1. Introduction

The Application Dictionary Services/400 (Feature 2212), a separately orderable feature of the Application Development ToolSet/400 (Program 5716-PW1) product, integrates a dictionary database program with other Application Development ToolSet/400 components. It stores descriptions of application objects and their relationships in an application dictionary, thereby maintaining a complete inventory of all objects. As you make changes to these objects, the information in the dictionary is automatically updated.

This version of Application Dictionary Services/400 (Version 3 Release 6) allows you to work with both Original Program Model (OPM) and Integrated Language Environment (ILE) applications.

This chapter describes:

- What ILE is
- The different user environments in which Application Dictionary Services/400 (AppDict Services/400) can be used
- What an application dictionary is
- The AppDict Services/400 functions

What is Integrated Language Environment?

In previous versions of Application Dictionary Services/400, the **Original Program Model (OPM)** was the only program model supported. The OPM is the traditional set of functions, processes, and rules provided by the Operating System/400 (OS/400) operating system to create and run a program. This version of AppDict Services/400 supports both OPM and ILE programs.

The **Integrated Language Environment (ILE)** is designed to enhance program development and maintenance on the AS/400 by allowing you to create programs that:

- Are modular
- Use multiple programming languages
- Reuse components from other programs, rather than duplicating them across programs
- Have better call performance
- Can be optimized to run faster

In addition to the benefits just described, converting your OPM programs to ILE creates a foundation for the object-oriented environments of the future.

Integrated Language Environment (ILE) support is new to Version 3 Release 6 of AppDict Services/400.

Description of the ILE Program Objects

In ILE, a few additional program units have been introduced. Programmers can still use OPM programs, but they can combine these programs with other program units that are unique to the ILE environment:

- ILE programs
- Service programs
- Modules

In ILE, all of the program parts that are going to be used by a larger application are created separately, and then they are bound together by the CRTPGM (Create Program) or CRTSRVPGM (Create Service Program) commands. The ILE environment does not allow the program parts to be run as individual units, even if their logic is self-contained. Therefore, modules cannot be run individually. However, service programs also cannot be run individually, since they contain a collection of separate procedures and data items.

Note: AppDict Services/400 does not support the creation of programs using the CRTBNDxxx command. It is recommended that if you have any single-module programs, you create them with the appropriate CRTxxxMOD command, followed by the CRTPGM command.

ILE Programs

An **ILE program** is a set of program units that have been bound together to create an executable program. An ILE program shares the following characteristics with an OPM program:

- The program gets control through a **dynamic program call**, which initiates and activates the called program at runtime
- The entry point to the program is the **program entry procedure (PEP)**, which is the compiler-generated code created for a dynamic program call
- The program is identified to the system by the symbol *PGM (PGM, if you are using AppDict Services/400 from within an Application Development Manager/400 session)

An ILE program has the following characteristics that an OPM program does not have:

- An ILE program is created from one or more module objects. For more information about modules, refer to “Modules” on page 3.
- One or more of the copied modules can contain a PEP.
- You have control over which module contains the PEP.

Also, procedures and data items *cannot* be exported from an ILE program, but procedures or data items *can* be imported from other ILE objects. An ILE program *can* be run as an individual unit.

Service Programs

A **service program** is a collection of procedures, which can be separately called when bound to another ILE program or service program, and available data items. A service program is represented to the system by the symbol *SRVPGM (SRVPGM, if you are using AppDict Services/400 from within an Application Development Manager/400 session). Because of the overhead required the first time a service program is called, service programs should contain common services that many other ILE program objects may need. For example, if a date conversion routine is used by many programs within a larger application, it should be placed in a service program; if it is only used in a couple of programs, it should be placed in a module that can be bound to the applications that need it.

The **public interface** to a service program consists of the names of the exported procedures and data items that other ILE objects need to access. The list of procedures and data items that can be exported is defined using **binder language**. A service program can be updated without having to re-create other ILE programs or service programs as long as the export list remains intact or new procedure names or data items are added to the end of the export list.

Service programs cannot be dynamically called, but can be **called by reference**. Calls to service programs are not made using a dynamic call. Instead, programs that call specific service programs are identified as a parameter of the CRTSRVPGM (Create Service Program) command. The system performs some binding to the service program at bind time, but does not copy any runnable code from the service program into the calling program as would be done for a dynamic program call.

Modules

A **module** is the unit created by an ILE compiler, but it cannot be run. (An OPM compiler produces a program that can be run.) A module is the basic building block for creating ILE programs, and a module is represented to the system by the symbol *MODULE (MODULE, if you are using AppDict Services/400 from within an Application Development Manager/400 session). A module can consist of one or more procedures and data item specifications, and these procedures and data items can be directly accessed by other ILE objects. Modules can also export and import procedures and data items.

What Is the Difference Between a Module and a Procedure?

The terms module and **procedure** are virtually synonymous from an RPG and a CL perspective. One source member in CL or RPG creates one module, which is called a procedure when it runs. Other languages may support modules that have multiple procedures. Therefore, the terminology used in ILE is:

- You create or change modules.
- You call or run procedures.

The Application Dictionary Services/400 User Environments

You can use AppDict Services/400 as a stand-alone tool, or you can use it from within an Application Development Manager/400 session. You choose which environment you want to use when you start AppDict Services/400.

If you are using AppDict Services/400 as a stand-alone tool, the functions available to you are introduced in “Overview of the Application Dictionary Services/400 Functions” on page 6.

Working with AppDict Services/400 in an Application Development Manager/400 session allows you to access your applications in terms of projects, groups, and parts, which Application Development Manager/400 supports. For more information about the Application Development Manager/400 functions that AppDict Services/400 supports, refer to “Supported Application Development Manager/400 Functions” on page 7.

For more information about starting AppDict Services/400 as a stand-alone tool, refer to “Using the STRADS Command” on page 10, and for more information about starting AppDict Services/400 in an Application Development Manager/400 session, refer to “Using the STRADS Command in an Application Development Manager/400 Session” on page 12.

What is an Application Dictionary?

An AppDict Services/400 application **dictionary** documents information about every object in your application's libraries, and saves it in a set of database files. This information is updated automatically every time you make changes to application objects, and it allows you to navigate through the application. A dictionary created using the AppDict Services/400 feature can document up to 99 libraries depending on their size. Note that the dictionary only contains information describing the objects; the objects, themselves, remain in the AS/400 libraries in which they were created.

What Is Automatic Refreshing?

Automatic refreshing of each dictionary is accomplished through a journal that records all system activities. As you create, delete, move, rename, save, restore, change, and copy objects, a special program keeps track of the journal entries and updates the information about each object in all of the dictionaries that document it.

Because the program reads the entries in the journal, and runs corresponding actions in sequence, there may be a delay before your actions appear on the listing displays. For example, when you create an object, you will not see it on the listing display until the documenting job is finished. Thus, it is recommended that you perform certain activities, such as creating new dictionaries, when the system load is relatively light.

The following dictionary actions affect automatic refreshing:

- Renaming or moving an object causes the object information in the dictionary to change as well.

- Deleting an object from outside AppDict Services/400 deletes the object, but not the object information in the dictionary. To remove the object information in the dictionary, you must specify option 4 (Delete).

Source members are only supported, and hence automatically refreshed, if they are updated within AppDict Services/400; other objects are supported both within and outside the feature.

Note: It is not recommended that you document programs that have been created with the CRTBNDxxx command. Because the CRTBNDxxx command creates a temporary module which it deletes after it creates the bound unit, this command is not supported in Application Dictionary Services/400. For more information about the commands that AppDict Services/400 uses to create programs, refer to “Creating Programs, Service Programs, and Modules in Application Dictionary Services/400” on page 40.

Types of Information Stored in an Application Dictionary

The two types of information stored and maintained in an application dictionary are:

- Static
- Derived

Static Information

The **static information** is extracted from the cross-reference information that the Operating System/400 (OS/400) maintains for each object. Static information includes:

- Information about field definitions
- Object descriptions
- Information about which physical files are accessed by a logical file
- Information about which fields are in a record format
- Information about Structured Query Language (SQL) tables, SQL views, and physical files accessed by an SQL view
- Information about which files, data areas, and other programs are used by a program

The amount of information that AppDict Services/400 can provide for your application will vary depending on the high-level languages and utilities you used to create your application objects. These languages and utilities provide varying amounts of the cross-reference information that AppDict Services/400 uses. (Refer to the *DB2 for OS/400 Database Programming* guide for a complete list of products and the amount of cross-reference information they support.)

Derived Information

The **derived information** is generated by the AppDict Services/400 feature, itself, and is concerned with those aspects of an object that are not part of its inherent description. It includes details on:

- Logical files based on a physical file
- SQL views based on an SQL view table
- The impact of a *file* change on files, programs, service programs, and modules
- The impact of a *field* change on files, programs, service programs, and modules

How Application Dictionary Services/400 Maintains Information in a Dictionary

The term **meta data** is used to refer to both kinds of information that are stored in an application dictionary—static and derived.

For Deleted Objects or Parts

When you delete an object that is documented in a dictionary, you need to indicate whether the meta data of this object is to be deleted as well. Sometimes it may be advantageous to leave the meta data in the dictionary, so that you can recreate the object easily if its source still exists.

The AppDict Services/400 uses the meta data differently when deleting objects or parts, depending on whether you are using the AppDict Services/400 as a stand-alone tool or from within an Application Development Manager/400 session:

- If you are using AppDict Services/400 as a stand-alone tool, and if only the meta data for an object exists, the AppDict Services/400 displays will indicate that the object does not exist in the system
- If you are using AppDict Services/400 in an Application Development Manager/400 session, only the parts that exist are displayed, regardless of whether or not the meta data exists

In both environments, as long as an object or part still exists on the system, you can delete its meta data.

For Created Objects or Parts

When you create a new object, the meta data for that object is updated in the dictionary as the object is created.

The AppDict Services/400 uses the meta data differently when creating objects or parts, depending on whether you are using the AppDict Services/400 as a stand-alone tool or from within an Application Development Manager/400 session:

- If you are using AppDict Services/400 as a stand-alone tool, as the object or part is being created, the name of the object will appear on the appropriate AppDict Services/400 list display, but its description will indicate that it does not yet exist in the system.
- If you are using AppDict Services/400 from within an Application Development Manager/400 session, the object or part will appear on the display only after it is successfully built.

Overview of the Application Dictionary Services/400 Functions

The AppDict Services/400 feature can help you design and maintain your applications by:

- Determining where fields are used in the objects that constitute your application, and to determine what files, programs, service programs, or modules would be affected by a change to a given field.

This enables you to access the impact of a change before it is made. For more information, refer to “Determining the Field Reference Hierarchy” on page 69.

- Ensuring that a change to a field is reflected in all of the objects that refer to that field, and if necessary, to recompile or recreate files, programs, service programs, or modules that are affected by this change.

For more information, refer to “Recreating Impacted Objects” on page 81.

- Identifying certain physical files as field reference files.

For more information, refer to Chapter 6, “Marking Field Reference Files” on page 65.

- Searching for strings in a source member, and scanning for externally described field names in Report Program Generator (RPG) programs to determine if a change to that field has real impact.

For more information, refer to “How to Find Strings and Scan RPG Source” on page 87.

- Determining which OPM or ILE programs call a specific program, or which OPM programs are called by a specific program; and determining which ILE programs or service programs bind a specific program, or which OPM programs, service programs, or modules are bound by a specific ILE program.

For more information, refer to “Determining the Program Calling Structure” on page 90.

- Creating physical files based on field reference files.

For more information, refer to “Creating a Field Reference File From a Template” on page 47.

- Building SQL table files based on field reference files, and building SQL index files based on the SQL table files and physical files.

For more information, refer to “Creating SQL Files” on page 57.

- Finding program objects through their relationships to other program objects.

For more information, refer to Chapter 8, “Finding Objects and Their Relationships to Other Objects” on page 87.

- Creating programs, service programs, and modules, and creating other types of program objects.

For more information, refer to “Creating Programs, Service Programs, and Modules in Application Dictionary Services/400” on page 40, “Creating Files” on page 46, and “Creating Other Types of Objects” on page 61.

Supported Application Development Manager/400 Functions

The AppDict Services/400 feature can also be used from within an Application Development Manager/400 session. When you are using AppDict Services/400 as a stand-alone tool, you document information about objects that reside in libraries; when you use AppDict Services/400 from within an Application Development Manager/400 session, you document information about parts that reside in groups. All the functions described in the previous section are available in an Application Development Manager/400 session except for:

- Creating physical files based on field reference files
- Building SQL table files based on field reference files, and building SQL index files based on the SQL table files and physical files

AppDict Services/400 provides the Application Development Manager/400 project and group relationships within the dictionary. Also supported are the following Application Development Manager/400 functions:

- Building a part
- Changing a part
- Changing part information
- Checking in a part
- Checking out a part
- Displaying a part
- Displaying part information
- Exporting a part
- Promoting a part

| For more information about the Application Development Manager/400 feature, refer to the *ADTS/400: Application Development Manager/400 User's Guide*.

Chapter 2. Getting Started

Before you can use the Application Dictionary Services/400 (AppDict Services/400) feature, one or more application dictionaries must already exist. This chapter describes how to:

- Create your first dictionary
- Start the AppDict Services/400 feature directly or from within an Application Development Manager/400 session, after you have created a dictionary

The examples used throughout the procedures in this book are for illustration only. You will not be able to complete these examples unless you first create the source code, libraries, and dictionaries.

Installing Application Dictionary Services/400 and Transferring Dictionaries

Because AppDict Services/400 now supports Integrated Language Environment (ILE) languages, you cannot migrate the dictionaries you used in an earlier version of AppDict Services/400 (V2R1.1, V2R2, V2R3, or V3R1) for use in the V3R6 version of AppDict Services/400.

To install AppDict Services/400, follow these steps:

1. If you will want to transfer the dictionaries you are currently using to this release of AppDict Services/400, print a list of the current dictionaries being used in the previous version (and the libraries documented within those dictionaries) before installing the new version of AppDict Services/400:
 - a. To get to the Work with dictionaries display, on a command line, type the following, and press the Enter key:

```
WRKAPPDCT
```
 - b. In the *Opt* prompt beside the first dictionary (that you will want to rebuild in V3R6 of AppDict Services/400), type 8 (Work with documented libraries).

The Work with Documented Libraries display appears showing the libraries that are documented in the selected dictionary.
 - c. Press F21 (Print list) to print a record of the libraries documented in the selected dictionary.
 - d. Repeat steps 1b through 1c to record all of the libraries documented in the dictionaries you want to rebuild in this release (V3R6) of AppDict Services/400.
2. Use the DLTLCIPGM (Delete Licensed Program) command to delete the old version of AppDict Services/400.
3. Use the GO LICPGM (Go Licensed Program) command to go to the Work with Licensed Programs display.
4. Select option 11 (Install licensed programs) to install the V3R6 AppDict Services/400 feature (5716-PW1).

5. Create the first dictionary to be used in this release of AppDict Services/400 as described in “Creating Your First Dictionary.”
6. Using the record you made in step 1 on page 9, document the rest of the libraries and dictionaries you need from the previous release of AppDict Services/400. For more information, refer to Chapter 4, “Working with Dictionaries” on page 27.

Creating Your First Dictionary

When you first install AppDict Services/400, you need to create a dictionary before you can start AppDict Services/400. You can either turn an existing library into a dictionary, or you can create a new library to document existing libraries. However, it is recommended that you create a new and dedicated library for each dictionary.

Depending on what your needs are, you can create your first dictionary in a couple of different ways:

- To create your first dictionary from a new library using AppDict Services/400 as a stand-alone tool, refer to “Creating Dictionaries Directly in Application Dictionary Services/400” on page 28
- To create your first dictionary from a new library from within an Application Development Manager/400 session, refer to “Creating Dictionaries in an Application Development Manager/400 Session” on page 30

Starting the AppDict Services/400 Tool

This section describes how to start the AppDict Services/400 feature using either the STRADS command or the WRKAPPDCT command.

Note: If you have not created any dictionaries in AppDict Services/400 yet, you will not be able to work with any of your application objects. For more information, refer to the previous section “Creating Your First Dictionary.”

Using the STRADS Command

The STRADS command allows you to start the AppDict Services/400 feature. For more information about the syntax of the STRADS command, refer to Appendix A, “Control Language Commands in Application Dictionary Services/400” on page 125.

The following steps show you how to use the STRADS command.

1. Type STRADS on the command line, and press F4 (Prompt).
The Start AppDict Services/400 (STRADS) display appears.

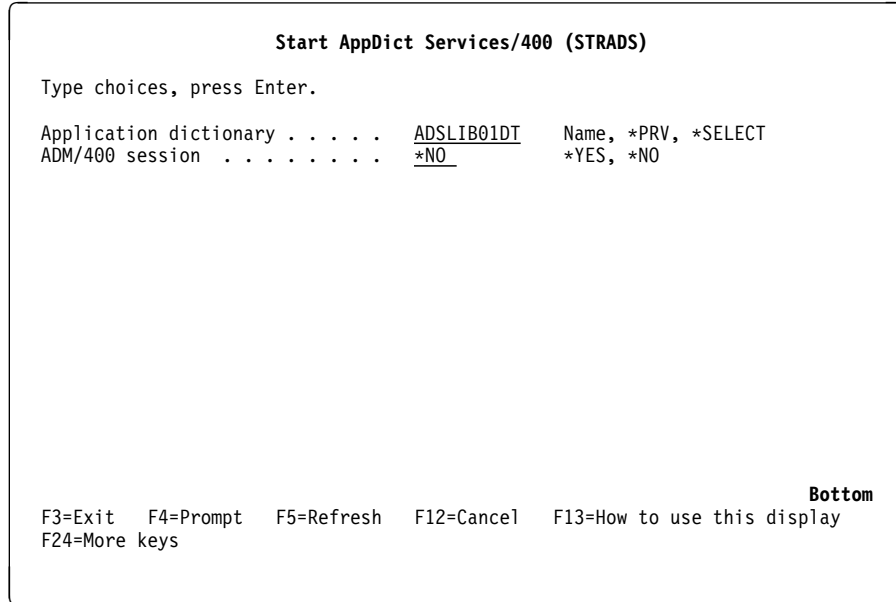


Figure 1. Start AppDict Services/400 (STRADS) Display

2. Fill in the prompts as follows, and press the Enter key:

a. In the *Application dictionary* prompt, specify the dictionary with which you want to work. You can specify:

- The name of a specific dictionary
- *PRV, to get the dictionary that was used in the previous session (the default)
- *SELECT, to show the Select Dictionary display which lists all of the available dictionaries from which you can choose the one you want.

In this example, type ADSLIB01DT.

b. In the *ADM/400 session* prompt, specify whether or not you want to work in an Application Development Manager/400 session.

In this example, specify *NO (the default).

The Application Dictionary Services/400 menu appears.

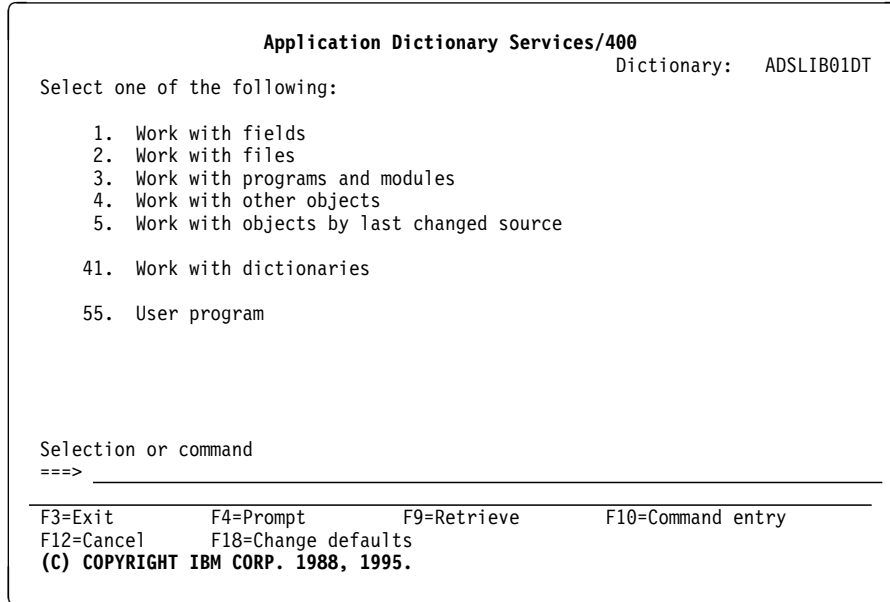


Figure 2. Application Dictionary Services/400 Menu

Using the STRADS Command in an Application Development Manager/400 Session

The following steps show you how to use the STRADS command to start AppDict Services/400 in an Application Development Manager/400 session.

1. Perform steps 1 through 3 as described in the previous section, except this time type *YES in the *ADM/400 session* prompt, and press the Enter key.

The Start AppDict Services/400 (STRADS) display for an Application Development Manager/400 session appears.

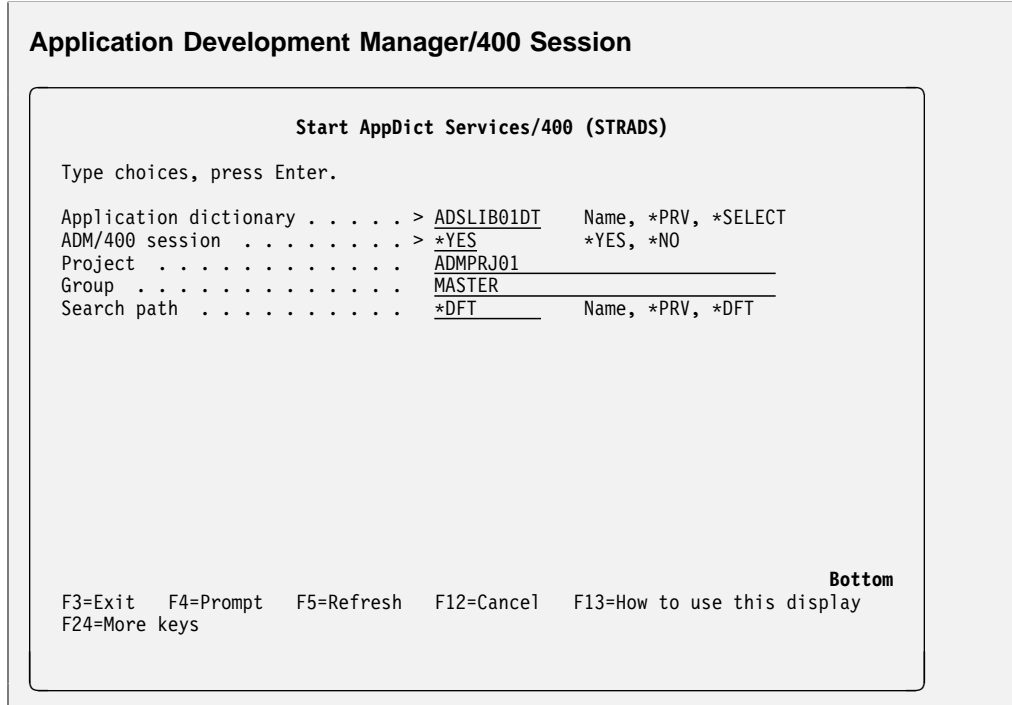


Figure 3. Start AppDict Services/400 (STRADS) Display in an Application Development Manager/400 Session

2. Fill in the prompts as follows, and press the Enter key:

- a. In the *Project* prompt, specify the project with which you want to work. You can specify:
 - The name of a specific project
 - *PRV, to get the project that was used in the previous session (the default).

In this example, type ADMPRJ01.

- b. In the *Group* prompt, specify the group name which will be your working group in an Application Development Manager/400 session.

Change activities will be performed in this group. This is also the group where AppDict Services/400 will begin looking for the SCHPTH part. If it is not found here, it will search from this group up the project hierarchy.

You can specify:

- The name of a specific group
- *PRV, to get the group that was used in the previous session (the default).

In this example, type MASTER.

- c. In the *Search path* prompt, specify a search path that will be used to subset the list and build the object.

For more information on how to work with search paths, refer to the *ADTS/400: Application Development Manager/400 User's Guide*.

You can specify:

- The name of a specific search path
- *PRV, to get the search path used in the previous session

- *DFT, to use QDFT as the search path (the default).

Note: If you use *DFT and no SCHPTH part called QDFT is found, the project hierarchy is searched.

In this example, type *DFT, and press the Enter key.

The Application Dictionary Services/400 menu appears.

Starting At the Application Dictionary Services/400 Menu

If you know the name of the dictionary with which you want to work (or dictionary, project, and group, for using AppDict Services/400 from within an Application Development Manager/400 session), you can use the STRADS command to go directly to the Application Dictionary Services/400 menu.

This section describes how to do this if you are using AppDict Services/400 as a stand-alone tool, or if you are using AppDict Services/400 from within an Application Development Manager/400 session. For more detailed information about the syntax of the STRADS command, refer to Appendix A, "Control Language Commands in Application Dictionary Services/400" on page 125.

Using Application Dictionary Services/400 As a Stand-Alone Tool

To bypass the Start AppDict Services/400 (STRADS) display and go directly to the Application Dictionary Services/400 menu to use AppDict Services/400 as a stand-alone tool, type the following, and press the Enter key:

```
STRADS dictionary-name
```

For example, you could type the following, and press the Enter key:

```
STRADS ADLIB01DT
```

Using Application Dictionary Services/400 in an Application Development Manager/400 Session

To bypass the Start AppDict Services/400 (STRADS) display and go directly to the Application Dictionary Services/400 menu in an Application Development Manager/400 session, type the STRADS command followed by the parameters you require, and press the Enter key. For example, you could type the following, and press the Enter key:

```
STRADS ADLIB01DT *YES ADMPRJ01 MASTER
```

Using the WRKAPPDCT Command

Instead of typing the STRADS command, you can also start the AppDict Services/400 feature by using the WRKAPPDCT command.

1. Type WRKAPPDCT on the command line, and press F4 (Prompt).

The Work with Dictionaries (WRKAPPDCT) display appears.

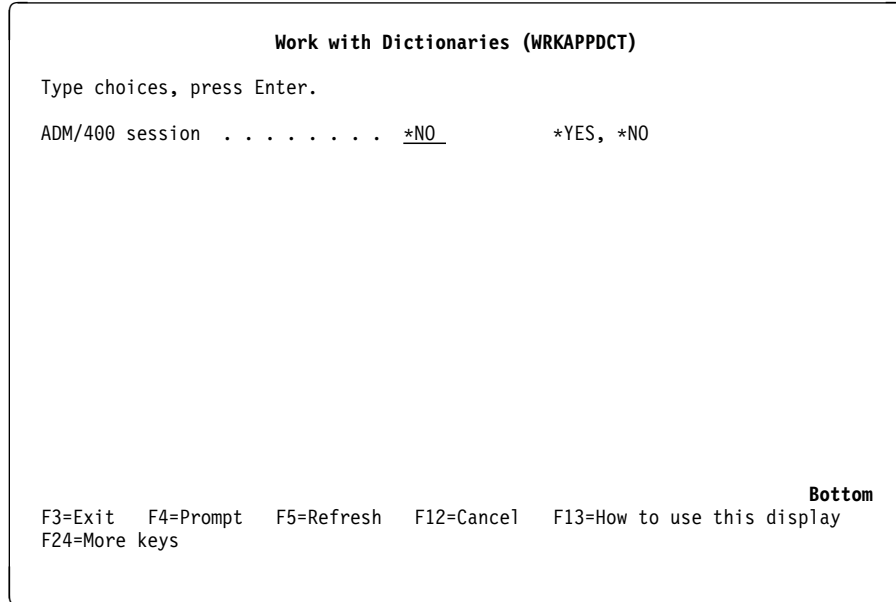


Figure 4. Work with Dictionaries (WRKAPPDCT) Display

2. Type *NO in the *ADM/400 session* prompt if you do not want to start an Application Development Manager/400 session, and *YES if you do. Type *NO in this case, and press the Enter key.

The Work with Dictionaries display appears.

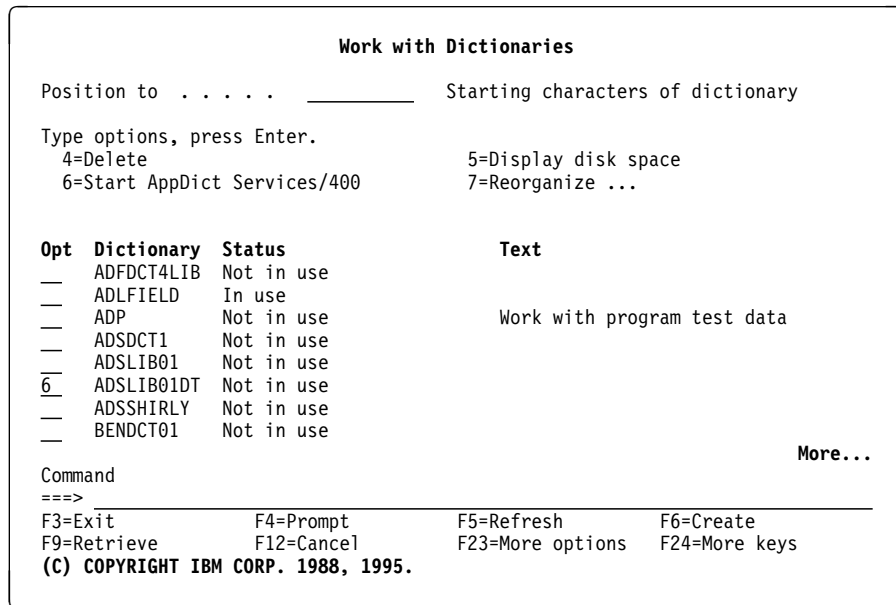


Figure 5. Work with Dictionaries Display. This example shows the dictionaries selected to work with.

3. Type 6 (Start AppDict Services/400) beside the dictionary with which you want to work (ADSLIB01DT in this case), and press the Enter key.

If you are using AppDict Services/400 as a stand-alone tool, the Application Dictionary Services/400 menu appears. If you are using AppDict Services/400 in an Application Development Manager/400 session, continue with step 4.

4. If you are in an Application Development Manager/400 session, the Start AppDict Services/400 (STRADS) display with more prompts appears. (Refer to Figure 3 on page 13.)
5. Specify the project, group, and search path name in the *Project*, *Group*, and *Search path* prompts, respectively, and press the Enter key.

The Application Dictionary Services/400 menu appears.

```

                                Application Dictionary Services/400
                                Dictionary:  ADSLIB01DT
Select one of the following:

    1. Work with fields
    2. Work with files
    3. Work with programs and modules
    4. Work with other objects
    5. Work with objects by last changed source

55. User program

Selection or command
====> _____

F3=Exit      F4=Prompt      F9=Retrieve      F10=Command entry
F12=Cancel   F18=Change defaults
(C) COPYRIGHT IBM CORP. 1988, 1995.
```

Figure 6. Application Dictionary Services/400 Menu

Note: If you want to start a debugging session before or after you start using the AppDict Services/400 tool, you need to specify *YES for the parameter *Update production files* when you use the STRDBG command.

Refreshing Your Screen

The AppDict Services/400 feature runs a background job for some functions on the list displays, such as copy and change, and updates the information about an object after the object has been copied or changed. Therefore, a refresh, F5 (Refresh), is only required to display the most current list.

Getting Help

| To get more information about any of the displays or prompts shown in AppDict
| Services/400, press F1 (Help). The help facility is context-sensitive, so a different
help panel appears depending on the position of the cursor when you press F1. To
scroll through displayed help panels, use the PageUp and PageDown keys.

If an option or a function key referred to in the text does not appear on your
display, press F23 (More options) or F24 (More keys), respectively.

Chapter 3. Making the Most of Your Application Dictionary Services/400 Working Environment

This chapter describes what you can do to enhance your Application Dictionary Services/400 (AppDict Services/400) working environment. You can:

- Change the defaults for the AppDict Services/400 functions
- Create your own AppDict Services/400 options
- Run multiple AppDict Services/400 sessions
- Work with a variety of AppDict Services/400 subsystems

You can use the functions described in this chapter to make your program environment easier to manage, especially if you are working with Integrated Language Environment (ILE) programs which contain substantially more parts than Original Program Model (OPM) programs.

For example, since ILE modules need to be compiled before the ILE programs and service programs that use them can be created or recreated, it would be useful to set the *Compile after editing* prompt to Y as described in the next section “Changing Function Defaults.”

Changing Function Defaults

You can change the defaults that control how the AppDict Services/400 functions are performed. You can change the:

- Values that control the compilation action to be taken after an object has been edited
- Job description
- User-defined option file
- User program that is started when you select option 55 (User program) from the Application Dictionary Services/400 menu

For example, you can set the defaults for your AppDict Services/400 feature to compile source members after you edit them, or you can specify the name of a user-defined option file instead of the default option file.

The following example shows you how to change the default function values:

1. From the Application Dictionary Services/400 menu, type 2 (Work with files) to get to the Subset Files to Work With display.
2. Specify the subsetting options needed, and press the Enter key to proceed to the Work with Files display.
3. Press F18 (Change defaults).

The Change Defaults display appears.

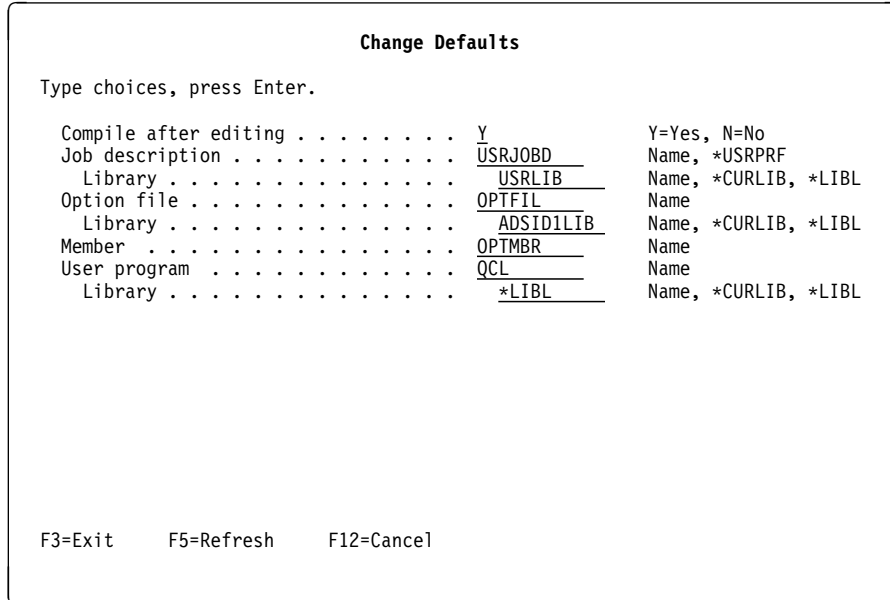


Figure 7. Change Defaults Display

You can do the following from this display:

- In the *Compile after editing* prompt, you can type Y to specify that source members are to be compiled after editing.
- In the *Job description* and *Library* prompts, you can type a new job description name (USRJOB in this case) and library name (USRLIB in this case) to change the job description for submitting batch jobs.
- In the *Option file*, *Library*, and *Member* prompts, you can type a new file name (OPTFIL in this case), library name (ADSID1LIB in this case), and member name (OPTMBR in this case), respectively, to change the name of the current user-defined option file.
- In the *User program* and *Library* prompts, you can type the name and location of a user program (QCL in this case) that is to be run when you select option 55 (User program) from the Application Dictionary Services/400 menu.

Creating Your Own Options

This section shows you, through examples, how you can use the AppDict Services/400 feature to create your own user-defined options for objects that appear on display lists, just as you can using the Programming Development Manager (PDM) tool.

You can turn any AS/400 system or user command into a user-defined option. User-defined options can contain substitution variables so that they can be used with any item in a display list.

Table 1 describes the permitted substitution variables you can use and the values returned for each one. Figure 8 on page 23 shows some examples of commands that use substitution variables.

Note: The variables beginning with Z are meant to be used only in an Application Development Manager/400 session. However, they can also be used when you are using Application Dictionary Services/400 as a stand-alone tool. For example, substitution variable &ZN can be used as an alternative name for items. To illustrate this, in the Work with Fields display using &ZN will return the file name and using &N will return the field name.

Table 1 (Page 1 of 2). Substitution values recognized by AppDict Services/400

Substitution Variable	Meaning	Description
&A	Object attribute	If you are working with objects, &A is used as the attribute type for an object on an object list.
&G	Job description library	&G is used as the value for the job description library on the Change Defaults display.
&H	Job description name	The value in &H is used as the job description value on the Change Defaults display.
&J	Job description	&J is used as the job description value on the Change Defaults display. Type the value for this variable in the format <i>library/job description</i> .
&L	Library name	If you are working with objects, the value in &L is used as the name of the library that contains these objects.
&MD	Dictionary name	The value in &MD is used as the the name of the library containing the dictionary with which you are working.
&MF	Source file name	The value in &MF is used as the the source file name for the list item beside which the user-defined option was typed.
&MG	Job description library	The value in &MG is used as the job description library value used to create the item in the list beside which the user-defined option was typed.
&MH	Job description name	The value in &MH is used as the job description value used to create the item in the list beside which the user-defined option was typed.
&MI	Field name	The value in &MI is used as the name of the field in the list beside which the user-defined option was typed.
&MJ	Job description	The value in &MJ is used as the job description value used to create the list item beside which the user-defined option was typed. Type in the value of this variable in the format <i>library/job description</i> .
&ML	Source library name	The value in &ML is used as the library name of the source file for the list item beside which the user-defined option was typed.
&MM	Source member name	The value in &MM is used as the source member name for the list item beside which the user-defined option was typed.
&MR	Record name	The value in &MR is used as the name of the record for the list item beside which the user-defined option was typed.
&MU	User profile name	The value in &MU is used as the user profile name used to create the list item beside which the user-defined option was typed.
&N	Item name	The value in &N is used as the name of the list item beside which the user-defined option was typed.

Table 1 (Page 2 of 2). Substitution values recognized by AppDict Services/400

Substitution Variable	Meaning	Description
&S	Item type without *	If you are working with objects, the value in &S is used as the object type without the *
&T	Item type with *	If you are working with objects, the value in &T is used as the object type as it is (with the *).
&U	User-defined option file	The value in &U is used as the user-defined option file name on the Change Defaults display
&V	User-defined option library	The value in &V is used as the user-defined option library name on the Change Defaults display
&W	User-defined option file member	The value in &W is used as the user-defined option file member name on the Change Defaults display
&X	Item text	The value in &X is used as the text (in single quotation marks) of the item in the list beside which the option was typed
&ZA	Part language	The value in &ZA is used as the part language
&ZG	Working group name	The value in &ZG is used as the working group name
&ZL	Group name	The value in &ZL is used as the name of the group beside which the user-defined options were typed
&ZN	Part name	The value in &ZN is used in the name of the part in the list beside which the user-defined option was typed
&ZP	Project name	The value in &ZP is used as the name of the project
&ZS	Search path	The value in &ZS is used as the value of the SCHPTH keyword from the STRADS command. This value is the name of the search path part.
&ZT	Part type	The value in &ZT is used as the type of the part
&ZX	Part text	The value in &ZX is used as the text (in single quotation marks) of the part

The following example shows you how to create a user-defined option to run a Data File Utility (DFU) temporary program that updates a data file:

1. From the Application Dictionary Services/400 menu, type 2 (Work with files), and press the Enter key.

The Subset Files to Work With display appears.

2. Fill in the prompts as follows, and press the Enter key:

- a. In the *File* prompt, type the name of a file that you previously defined.
- b. In the other prompts, type *ALL.

The Work with Files display appears.

3. Press F16 (User options).

The Work with User-Defined Options display appears, listing all the user-defined options defined in the current user-defined options file (QDNOOPT in this case) of library QGPL.

To use user-defined options in a different member, file, or library, you must update the appropriate prompts on the Change Defaults display. See “Changing Function Defaults” on page 19 for more information.

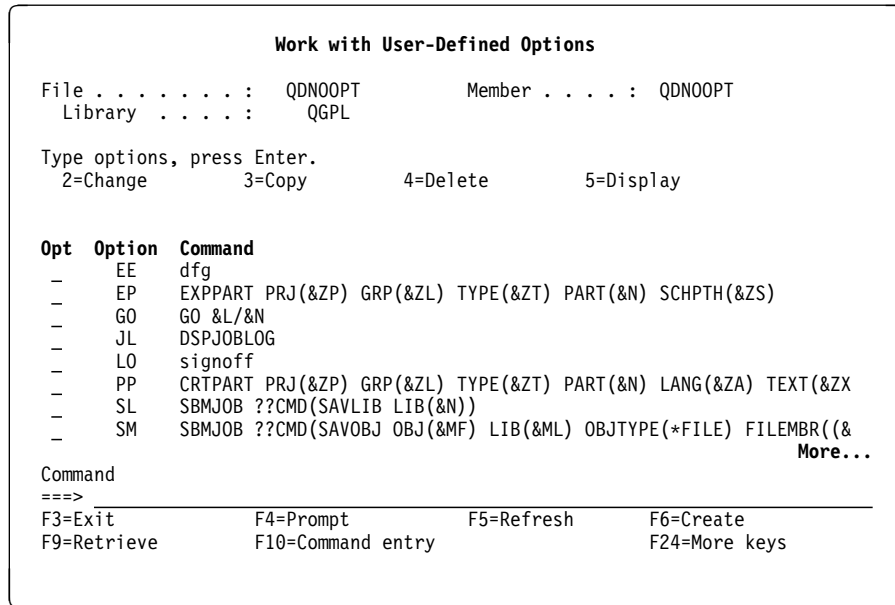


Figure 8. Work with User-Defined Options Display. This example shows the user-defined options available.

4. On the Work with User-Defined Options display, press F6 (Create).
The Create User-Defined Option display appears.

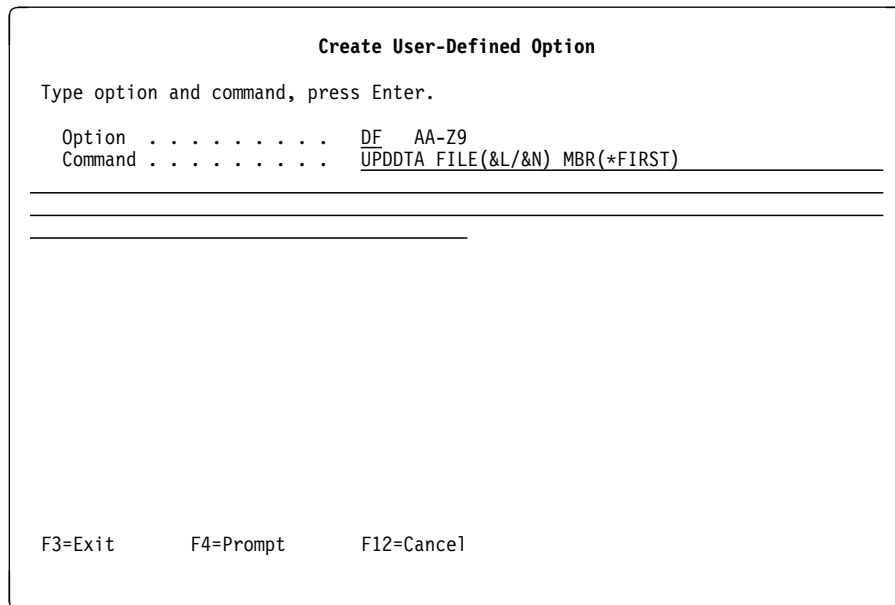


Figure 9. Create User-Defined Option Display

5. Fill in the prompts as follows, and press the Enter key:
 - a. In the *Option* prompt, type the name of the user-defined option (in this case, DF).

- b. In the *Command* prompt, type the DFU command to create and run a temporary program.

```
UPDDTA FILE(&L/&N) MBR(*FIRST)
```

The new user-defined option is created, and the Work with User-Defined Options display appears.

6. To return to the Work with Files display, press the Enter key again.
7. Type DF in the *Opt* prompt, and press the Enter key.

The DFU temporary program is run.

Running Multiple Sessions

Sometimes, you might want to be able to switch between related AppDict Services/400 tasks without having to end one before starting another. The AppDict Services/400 feature allows you to work with up to five sessions at the same time.

You can complete the following steps from anywhere within AppDict Services/400:

1. Start the AppDict Services/400 as described in “Using the STRADS Command” on page 10 or “Using the STRADS Command in an Application Development Manager/400 Session” on page 12.
2. Begin to work with any part of AppDict Services/400 that you want.
3. Press the attention key.

If you are using AppDict Services/400 in an emulator window, this key will be the one defined for your workstation in the keyboard mappings. If you are using a terminal, this key will vary in appearance, depending on the type of terminal you are using.

The Select Session menu appears.

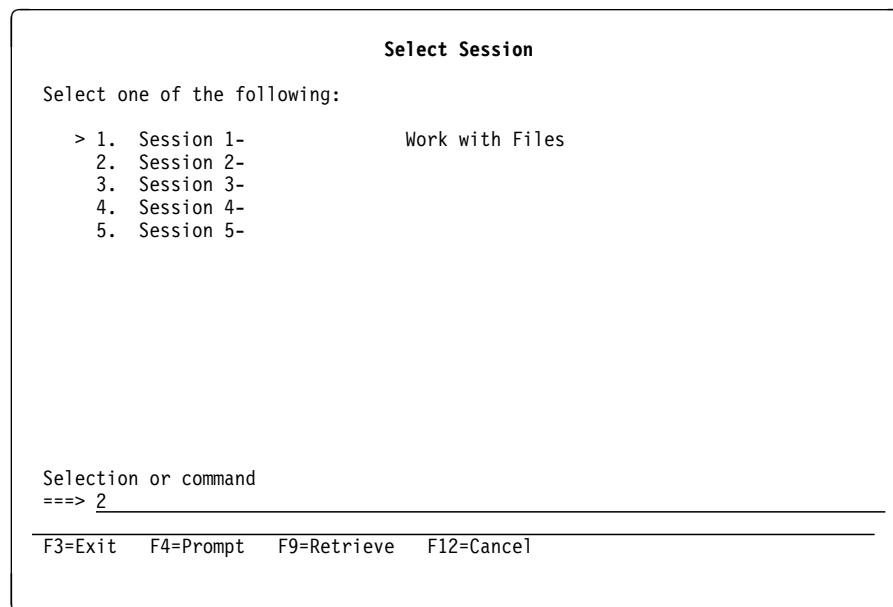


Figure 10. Select Session Menu

The current session that you are working with is indicated by the > symbol on the Select Session menu. In Figure 10, session 1 is assigned, while sessions 2 through 5 have not yet been assigned.

4. Select an option that corresponds to a session that has not yet been assigned. For example, type 2 for session 2, and press the Enter key.

The Application Dictionary Services/400 menu appears (see Figure 2 on page 12).

Note: If there is more than one session assigned, pressing F3 (Exit) on the Select Session menu will end the current session. If there is only one session assigned, pressing F3 (Exit) will take you to that session.

Figure 11 illustrates the concept of switching between sessions using the attention key.

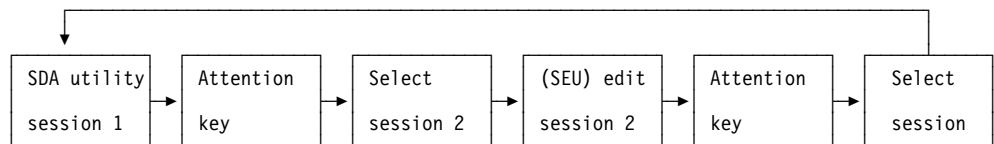


Figure 11. Example of Switching Tasks Using the Attention Key



Tip:

If you have set up the attention key to call another program and you have pressed this key before starting AppDict Services/400, the setting of this key will remain the same. However, if before setting up the attention key to call another program and before starting AppDict Services/400, you have pressed the attention key, the setting of your attention key will be changed. In this case, if you want to use the multiple sessions function, you must sign off your AS/400 system, then sign on again; and you must not press the system attention key before starting AppDict Services/400.

Working with Subsystems

In the AppDict Services/400 feature, two subsystems are available:

- QDMT

This subsystem is used when you use the AppDict Services/400 options that create, update, and rebuild the dictionaries.

- QDNNOTIFY

QDNNOTIFY is used by the automatic refresh function to automatically rebuild and update the dictionaries.

The very first user to issue the STRADS or WRKAPPDCT command automatically starts these subsystems, and they continue to run even when this user signs off the AS/400 system.

It is possible to stop the QDNNOTIFY subsystem, but it is not recommended because it then may become necessary to rebuild your dictionaries. To stop the QDMT and QDNNOTIFY subsystems, either:

- After you have gotten all of the users to exit the AppDict Services/400 feature, issue the ENDSBS command to stop users from starting the AppDict Services/400 feature
- Remove the AppDict Services/400 feature from the system

You may want to hold the QDNNOTIFY subsystem so that other jobs in the system may run. To hold this subsystem, type option 3 (Hold) beside it in the Work with Active Jobs display. To release this subsystem, type option 6 (Release) beside it on the same display.

You can only hold the QDNDCTSYNC job when its status is RUN or TIMW.

Chapter 4. Working with Dictionaries

As you start and end development work on products, you will need to plan which libraries and dictionaries you are going to use. If you are starting work on a product that is completely different from anything else you have done, or you are starting a new release of an existing product, you will probably want to set up new dictionaries and libraries.

As your product development progresses, you will also need to maintain your existing dictionaries to meet any changes in work flow that may arise, and to keep your development environment organized.

This chapter describes how to:

- The statuses that dictionaries can have
- How to:
 - Create dictionaries
 - Add or remove the documentation of libraries and groups from dictionaries
 - Reorganize dictionaries
 - Delete dictionaries
 - Find documented libraries and documenting dictionaries
 - Support multiple projects with one dictionary

Most often, all of the activities you complete in the Application Dictionary Services/400 (AppDict Services/400) feature are recorded in a journal and are automatically processed by the affected dictionaries. This is called **automatic refreshing**. Occasionally, it might be necessary to check whether or not all of the journal entries have been processed, and take the appropriate action if they have not. For more information about automatic refreshing and how to check the journal entries, refer to “Ensuring Your Dictionaries Are Automatically Refreshing” on page 119 and “Determining Whether or Not Automatic Refreshing Is Working” on page 120.

Note: It is *not recommended* that you document programs that have been created with the CRTBNDxxx command. Because the CRTBNDxxx command creates a temporary module which it deletes after it creates the bound unit, this command is not supported in AppDict Services/400. For more information about the commands that AppDict Services/400 uses to create programs, refer to “Creating Programs, Service Programs, and Modules in Application Dictionary Services/400” on page 40.

Statuses of Dictionaries

The following are the possible values in the *Status* prompt:

Status	Description
In use	Indicates that some interactive or batch jobs are currently accessing the dictionary.
Not in use	Indicates that no jobs are currently accessing the dictionary.
Processing	Indicates that a dictionary creation, or reorganization job is currently in process.

Not complete Indicates that a job that was submitted to create or rebuild the dictionary:

- Failed
- Was cancelled by the user
- Has been waiting in the message queue for more than 24 hours



Tips:

1. If the job documenting a dictionary is cancelled or if it fails, the system will immediately process the job that is next in the queue, and the status of the failed dictionary will become Not complete. The next time you start AppDict Services/400 with this dictionary, a documentation job will be submitted to document the dictionary from last checkpoint, and the status of the dictionary will then become Processing.
 2. If the job documenting a dictionary takes a long time (more than 24 hours), the status of the queued job becomes Not complete. In this case, you do not have to restart it. It will automatically start documenting. Once the first job finishes documenting, the next job, if one exists, will automatically start documenting.
-

Creating Dictionaries

It is recommended that you start with an empty and dedicated library for each new dictionary.

This section describes how to create:

- Dictionaries directly in AppDict Services/400
- Dictionaries from within an Application Development Manager/400 session
- Multiple dictionaries

Creating Dictionaries Directly in Application Dictionary Services/400

In this example, we will create an empty and dedicated library called ADSLIB01DT, and turn it into a dictionary by documenting the contents of an existing library in it.

Note: A dictionary name can be up to 10 characters. It must begin with an alphabetic character (A through Z, a through z, \$, #, or @), and can be followed by up to 9 alphanumeric characters (A through Z, a through z, 0 through 9, \$, #, @, period (.), or underline (_)). For example, you could have a name such as ADS01LIBDT. Blanks and quotation marks are not allowed. For example, it could not be "Mydict."

1. Create a new library with the CRTLIB command. Type the following, and press the Enter key:

```
CRTLIB library-name
```

For example, you could type the following, and press the Enter key:

```
CRTLIB ADSLIB01DT
```

The library you specified is created.

Note: The user creating this library is given *ALL authority to it. To use this library you must have at least *CHANGE authority to it.

2. Type the STRADS command, followed by the name of the library you want to turn into a dictionary, and press the Enter key:

```
STRADS library-name
```

For example, you could type the following, and press the Enter key:

```
STRADS ADSLIB01DT
```

The Document Libraries for Dictionary display within the AppDict Services/400 feature appears. Figure 12 shows the library to be added (AP01.MST, in this case).

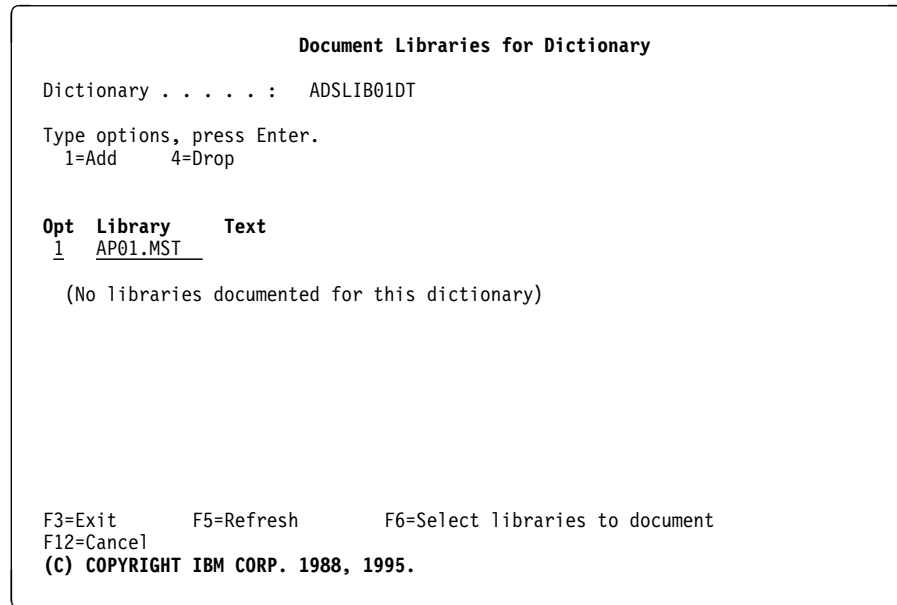


Figure 12. Document Libraries for Dictionary Display. This example shows the library to be added.

3. Fill in the prompts as follows, and press the Enter key:

- a. In the *Opt* prompt, type 1 (Add).
- b. In the *Library* prompt, type the name of the library you want to document (AP01.MST in this case).

If you do not remember the name of the library you want, press F6 (Select libraries to document) to view a list of all of the libraries, and then select one or more.

If you change your mind and decide not to document a library, drop its entry by typing option 4 (Drop) beside its name, and press the Enter key.

The dictionary creation job is submitted to the batch system.

You are returned to whatever display you were on when you typed the STRADS command. A message appears to tell you that the dictionary creation job has been submitted.



Tip:

When you submit the batch job, a number of less important batch jobs that document the individual source member are automatically submitted as well. All these jobs take some time to process. The name of the new dictionary is immediately added to the Work with Dictionaries display (shown in Figure 5 on page 15); however, it is not immediately available to work with. To determine when it is available, check the *Status* column on that display. If the dictionary is still being created, its status will be *Processing*. If the dictionary is completed but the source members are still being created, its status will be *In use*. When the status has changed to *Not in use*, you can begin to work with the dictionary.

4. To confirm that the dictionary has been created, type the `DSPMSG` command to look at the system messages.

Creating Dictionaries in an Application Development Manager/400 Session

When you use the AppDict Services/400 feature from within an Application Development Manager/400 session, the AppDict Services/400 feature will turn a new library into a dictionary when information from groups is documented into the library.

In this example, we will create an empty and dedicated library called `ADSLIB01DT` from within an Application Development Manager/400 session, and turn it into a dictionary by documenting the contents of the existing groups in it.

1. Create a new library with the `CRTLIB` command. The following is the format of the Create Library (`CRTLIB`) command:

```
CRTLIB library-name
```

For example, type the following, and press the Enter key:

```
CRTLIB ADSLIB01DT
```

The library `ADSLIB01DT` is created.

The user creating this library (you, in this case) is given `*ALL` authority to it. To use this library you must have at least `*CHANGE` authority to it.

2. Type the `STRADS` command, followed by the parameters described below, and then press the Enter key:
 - a. The name of the application dictionary to be created (`ADSLIB01DT` in this case).
 - b. Whether or not this is an Application Development Manager/400 session (`*YES` in this case).
 - c. The name of the project from which groups will be documented (`ADMPRJ01` in this case).

Thus, in this example you would type the following, and press the Enter key:

```
STRADS ADSLIB01DT *YES ADMPRJ01
```

The Work with Documented Groups display appears.

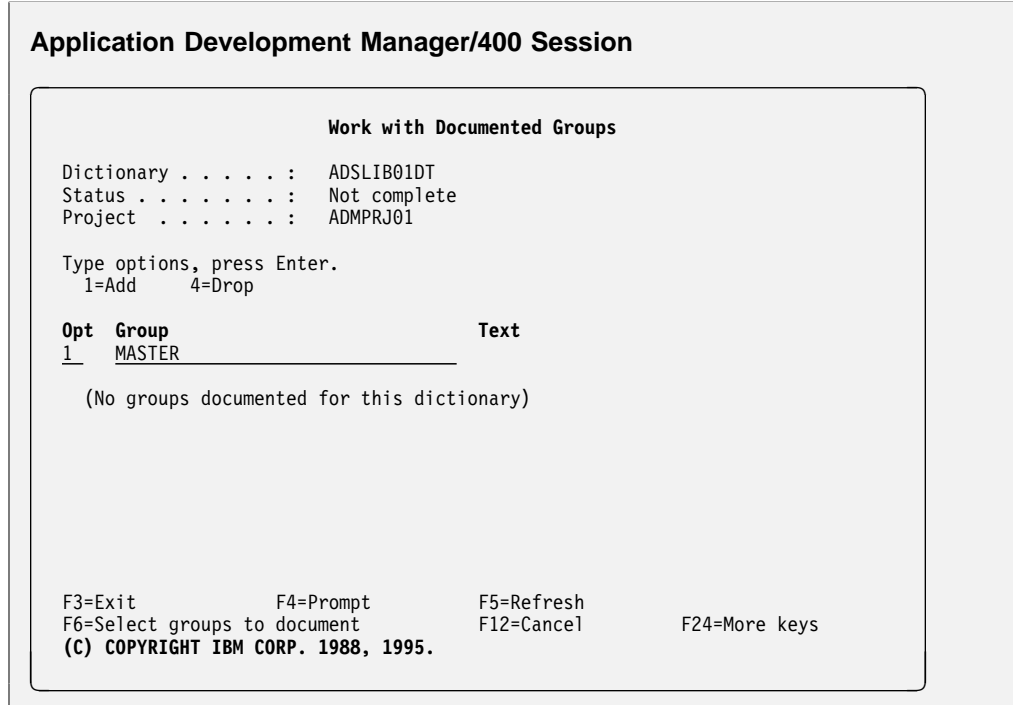


Figure 13. Work with Documented Groups Display in an Application Development Manager/400 session. This example shows the groups to be added.

3. Fill in the prompts as follows, and press the Enter key:

- a. In the *Opt* prompt, type 1 (Add).
- b. In the *Group* prompt, type the name of the group you want to document (MASTER in this case).

If you do not remember the name of the group that you want, press F6 (Select groups to document) to view a list of all of the groups in the project.

If you change your mind and decide not to document a group, drop its entry by typing option 4 (Drop) beside its name.

4. To submit the dictionary creation job to the batch system, press the Enter key.

You are returned to whatever display you were on when you typed the STRADS command. A message appears to tell you the dictionary creation job has been submitted.

5. To confirm that the dictionary has been created, type the DSPMSG command to see the system messages.

Creating Multiple Dictionaries

With AppDict Services/400, you can create multiple dictionaries at the same time. However, creating multiple dictionaries could take a while, because the dictionaries are created one at a time. It is recommended that you create multiple dictionaries after peak hours, when the load on the system is not as great.

Adding or Removing the Documentation of Libraries and Groups

From time to time, you might find it necessary to reorganize the structure of libraries documented within dictionaries. You might need to document additional libraries or groups in a dictionary, or remove libraries or groups from a dictionary.

Documenting Additional Libraries or Groups in a Dictionary

You may sometimes need to create additional libraries (or groups, if you are in an Application Development Manager/400 session) for your application.

Documenting Additional Libraries

To add libraries to a dictionary:

1. Type WRKAPPDCT on the command line, and press the Enter key.

The Work with Dictionaries display appears, showing the available dictionaries.

2. Type 8 (Work with documented libraries) beside the name of the dictionary to which you want to add libraries (ADSLIB01DT in this case), and press the Enter key.

The Work with Documented Libraries display appears, showing the libraries that are currently documented in this dictionary. In this example, the only library is AP01.MST.

```

                                Work with Documented Libraries
Dictionary . . . . . : ADSLIB01DT
Status . . . . . : Not in use
Position to . . . . . : _____ Starting characters of library

Type options, press Enter.
 4=Drop
 8=Display dictionaries documenting a library

Opt  Library  Text
___  AP01.MST

                                Bottom
F3=Exit      F5=Refresh      F6=Document library      F10=Command entry
F12=Cancel   F18=Change defaults      F21=Print list
```

Figure 14. Work with Documented Libraries Display. This example shows the libraries available in the dictionary.

3. Press F6 (Document library).

The Document Libraries for Dictionary display appears, with a prompt for the libraries you want to add to the dictionary.

Note: It is recommended that you do not document a **dictionary library** into a dictionary because the information about the dictionary is not relevant to your application, and this action adds only unnecessary complexity to the information provided in the dictionary. A dictionary library is the

empty dedicated library that you use to create a dictionary. AppDict Services/400 uses this dictionary library to store the information it keeps about your application objects.

4. Fill in the prompts as follows, and press the Enter key:
 - a. In the *Opt* prompt, type 1 (Add).
 - b. In the *Library* prompt, type the name of the library that you want to add to the dictionary ADSLIB01DT.

Note: If you do not remember the name of the library that you want to document, press F6 (Select libraries to document) to view a list of libraries from which you can select one or more.

The new library name is added to the list of libraries on the Document Libraries for Dictionary display.

5. Repeat steps 4a and 4b to add another library to the dictionary.
6. To submit a batch job to add the libraries, press the Enter key.

Documenting Additional Groups

To add groups to a dictionary in an Application Development Manager/400 session:

1. Type WRKAPPDCT ADM(*YES) on the command line, and press the Enter key.

The Work with Dictionaries display appears, showing the available dictionaries.
2. Type 8 (Work with documented groups) beside the dictionary to which you want to add groups (ADSLIB01DT in this case), and press the Enter key.

The Specify Project to Work With display appears.

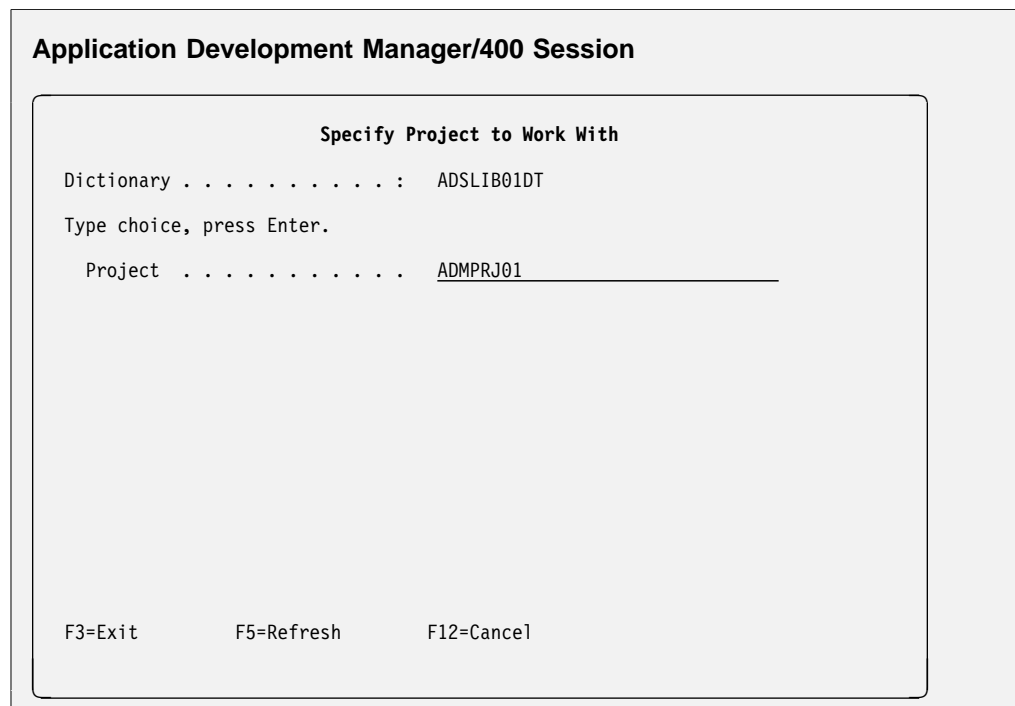


Figure 15. Specify Project to Work With Display in an Application Development Manager/400 Session

3. Specify the name of the project with which you want to work (ADMPRJ01 in this case), and press the Enter key.

The Work with Documented Groups display appears, showing the groups that are currently documented in this dictionary. Refer to Figure 13 on page 31.

4. Fill in the prompts as follows, and press the Enter key:
 - a. In the *Opt* prompt for the top list entry, type 1 (Add).
 - b. In the *Group* prompt, type the name of the group that you want to add to the dictionary ADSLIB01DT. If you do not remember the name of the group that you want to add, press F6 (Select groups to document) to view a list of groups and then select one or more.

The new group name is added to the list of groups on the Work with Documented Groups display.

5. Repeat steps 4a and 4b to add another group to the dictionary.
6. To submit a batch job to add the groups, press the Enter key.

Removing the Documentation of Libraries or Groups from a Dictionary

You may sometimes want to remove unnecessary documentation for libraries or groups from a dictionary.

Removing Documentation of Libraries

To remove the documentation of a library:

1. Type WRKAPPDCT on the command line, and press the Enter key.

The Work with Dictionaries display appears, showing the available dictionaries.

2. Type 8 (Work with documented libraries) beside a dictionary to list all its documented libraries, and press the Enter key.

The Work with Documented Libraries display appears, showing the libraries that are currently documented in this dictionary.

3. Type 4 (Drop) beside the libraries you want to remove, and press the Enter key.

The library is removed from the list of libraries displayed.

Note: You may remove more than one library in the same session.

4. To submit a batch job to remove the libraries from the dictionary, press the Enter key again.

Removing Documentation of Groups

To remove the documentation of a group in an Application Development Manager/400 session:

1. Type WRKAPPDCT ADM(*YES) on the command line, and press the Enter key.

The Work with Dictionaries display appears, showing the available dictionaries.

2. Type 8 (Work with documented groups) beside a dictionary, and press the Enter key.

The Specify Project to Work With display appears.

3. Specify the name of the project with which you want to work, and press the Enter key.

The Work with Documented Groups display appears, showing the groups that are currently documented in this dictionary.

4. Type 4 (Drop) beside the groups you want to remove, and press the Enter key.
The group is removed from the list of groups displayed.
Note: You may remove more than one group in the same session.
5. To submit a batch job to remove the groups from the dictionary, press the Enter key again.

Reorganizing a Dictionary

When a library or a group has been removed from a dictionary, AppDict Services/400 reuses the disk space. However, this disk space may become fragmented. Therefore, you may want to reorganize the dictionary to recover the unused disk space. Only a dictionary whose status is Not in use can be reorganized.

To reorganize a dictionary:

1. Type WRKAPPDCT on the command line, and press the Enter key.
The Work with Dictionaries display appears, showing the available dictionaries.
2. On the Work with Dictionaries display, type 5 (Display disk space) beside the dictionary you choose (ADSLIB01DT in this case), and press the Enter key.
The Display Disk Space of Dictionary display appears.

```

                                Display Disk Space of Dictionary
Dictionary . . . . . : ADSLIB01DT
Status . . . . . : Not in use
Date of last reorganization . . . . . : 01/12/95
Date of last disk space analysis . . . . . : 01/12/95
Disk space used in kilo bytes . . . . . : 2,137
Disk space recoverable through reorganization:
  In kilo bytes . . . . . : 0
  In percentage . . . . . : 0

Press Enter to continue.

F3=Exit      F6=Perform disk space analysis
F12=Cancel
```

Figure 16. Display Disk Space of Dictionary Display

This display shows the amount and the percentage of disk space that can be recovered if the dictionary is reorganized. The information that appears is based on the date of the last disk space analysis, as shown. The value shown in the *In kilo bytes* prompt does not include the size of the dictionary library.

3. To get an updated analysis, press F6 (Perform disk space analysis).
The analysis result appears.
4. To return to the Work with Dictionaries display, press the Enter key.

5. In the *Opt* prompt, type 7 (Reorganize) beside the dictionary, and press the Enter key to submit a batch job to reorganize the dictionary.

A message appears to tell you that the dictionary is being reorganized. After the status of the dictionary has changed from *Processing* to *Not in use* on the *Work with Dictionaries* display, you can verify that disk space has been recovered by repeating step 2.

Deleting a Dictionary

Only the dictionary whose status is *Not in use* or *Not complete* can be deleted.

To delete a dictionary:

1. Type `WRKAPPDCT` on the command line, and press the Enter key.

The *Work with Dictionaries* display appears, showing the available dictionaries.

2. Type option 4 (*Delete*) next to its name on the *Work with Dictionaries* display, and press the Enter key.

The *Confirm Delete of Dictionaries* display appears.

3. Press the Enter key again.

All of the dictionary objects are deleted from the dictionary library. However, the dictionary library remains in the system.

Finding Documented Libraries and Documenting Dictionaries

When you are working in *AppDict Services/400*, often you need to know which dictionary documents the libraries with which you want to work. For example, you can create program objects only in libraries that are documented in the dictionary in which you are working. Alternatively, you also might want to know which libraries are documented in the dictionary with which you want to work.

To find out which dictionaries document the library with which you want to work, or which libraries are documented by the dictionary with which you are working:

1. From the *Application Dictionary Services/400* menu, type 41 (*Work with dictionaries*).

The *Work with Dictionaries* display appears.

2. Beside the dictionary in question, type 8 (*Work with documented libraries*), and press the Enter key.

The *Work with Documented Libraries* display appears, showing the libraries that are documented in the specified dictionary.

3. To look at the other dictionaries that document any of the libraries in the list, type 8 (*Display dictionaries documenting a library*) beside the library you are interested in, and press the Enter key.

The *Display Documenting Dictionaries* display appears.

Supporting Multiple Projects with One Dictionary

When you are using AppDict Services/400 in an Application Development Manager/400 session, you can also set up multiple projects in one dictionary. For example, you might want to keep similar projects in one dictionary so that you can work with program parts in related projects more easily.

You can support multiple projects in a dictionary, as follows:

1. Start an AppDict Services/400 session using the WRKAPPDCT command, with the ADM parameter set to *YES. To do this, on the command line type the following, and press the Enter key:

```
WRKAPPDCT ADM(*YES)
```

The Work with Dictionaries display appears, showing the available dictionaries.

2. Press F6 (Create) to create a dictionary with one project.

The procedure for creating dictionaries for one project is described in “Creating Dictionaries” on page 28.

3. After this dictionary is created, add groups or drop groups to and from it.

4. Exit AppDict Services/400; then restart it using the WRKAPPDCT command, with the ADM parameter set to *NO. To restart AppDict Services/400, type the following, and press the Enter key:

```
WRKAPPDCT ADM(*NO)
```

The dictionary that you created in step 2 should be listed on your display with a Not in use status.

5. On the Work with Dictionaries display, type option 8 (Work with documented libraries) beside this dictionary name.

The Work With Documented Libraries display appears.

6. Press F6 (Document library).

The Document Libraries for Dictionary display appears, where you may add or drop libraries.

7. To add another project to the dictionary, type 1 in the *Opt* prompt, and type the combined name of the project and group in the *Library* prompt.

For example, to add a project called PROJ and a group called TEST to a dictionary, type 1 in the *Opt* prompt, and type PROJ.TEST in the *Library* prompt, and press the Enter key to submit a batch job that will update the dictionary with PROJ.TEST.

8. To see the updates in the dictionary, start a new session using the WRKAPPDCT command, with the ADM parameter set to *YES.

To do this, type the following, and press the Enter key:

```
WRKAPPDCT ADM(*YES)
```

9. Beside the dictionary name, type 8, and press the Enter key.

The Specify Project to Work With display appears.

10. Type the name of the documented project with which you want to work.

Chapter 5. Creating Program Objects

This chapter describes how:

- The AS/400 system creates programs, service programs, and modules
- Data items and procedures are accessed across ILE program objects
- In AppDict Services/400, you can create:
 - Programs, service programs, and modules
 - Files based on field reference files (which includes adding fields)
 - SQL Files
 - Other types of objects

How Does the AS/400 System Create Programs, Service Programs, and Modules?

In OPM, a program is created in a single step. The compiler generates the program object which contains additional code. This additional code initializes program variables and provides any necessary code for special processing that is required by the particular language. For example, special processing could include handling any input parameters expected by the program. When the program is started, or called, the additional compiler-generated code becomes the starting point (entry point) for the program.

The process of creating ILE programs involves two steps:

- Compiling the source code into modules.
- Binding the modules into an ILE program or service program.

Binding occurs when the Create Program (CRTPGM) or Create Service Program (CRTSRVPGM) command is run.

The CRTPGM and CRTSRVPGM commands bind:

- Service programs and modules by copy when they are specified in the MODULE keyword
- Service programs by reference (to other service programs or ILE programs) when they are specified in the BNDSRVPGM keyword



Tip:

If you use a Binding Directory (BNDDIR) with the CRTPGM command, the modules are bound by copy only if they provide exports for unresolved imports. For more information about imports and exports, refer to “Accessing Data Items and Procedures Across ILE Program Objects” on page 40.

Accessing Data Items and Procedures Across ILE Program Objects

In OPM, support for sharing data between programs in a larger application is limited. Typically, to share data between programs in an application, data has to be passed as parameters to a CALL statement. This method is normally quite effective, except in those cases where the data is not processed by the next program in sequence.

For example, suppose that program A originates a piece of data that is to be processed by program D. If the normal sequence of events is for A to call B, which calls C, which calls D, the parameter has to be passed from program to program, even though neither B nor C uses it.

When you create an ILE program or service program object, you need to specify how other programs can access that program. On the CRTPGM command, you do so with the Entry Module (ENTMOD) parameter. On the CRTSRVPGM command, you do so with the Export (EXPORT) parameter.

When you create a program, service program, or module in AppDict Services/400, the appropriate system display appears for the create command that AppDict Services/400 is using. These system displays allow you to enter other parameters, such as ENTMOD.

The ENTMOD parameter specifies the module that contains the PEP and **User Entry Procedure (UEP)**. The UEP is the procedure (written by the programmer) that gets control when a dynamic program call is made; in contrast, the PEP is the compiler-written code that is the entry point to an ILE program when a dynamic program call is made.

The EXPORT parameter (along with the Source File (SRCFILE) and Source Member (SRCMBR) parameters) identifies the **public interface** to the service program being created. The public interface defines the data items and procedures that a service program makes available (exports) to other ILE programs or service programs.

Creating Programs, Service Programs, and Modules in Application Dictionary Services/400

Notes:

1. Remember that your modules have to be compiled before a properly bound program can be created from them.
2. If you are creating an SQL service program, you must already have a valid export list with a member name matching the name of the program.

You can use the AppDict Services/400 feature to create OPM and ILE programs. For OPM programs, however, the source file must exist before you can do so. For ILE programs and service programs, the modules and service programs that are bound to create the target object must exist.

In AppDict Services/400, a service program is created if you specify *SRVPGM as the type, and the data items and procedures that can be used by other programs are defined in the EXPORT parameter. An OPM program or ILE program is created if you specify *PGM as the type, and the PEP (the entry point to the ILE or OPM program on a dynamic call) is defined in the ENTMOD parameter. Both the EXPORT and ENTMOD parameters can be prompted from the Create Program, Service Program, or Module display. (If you are using AppDict Services/400 from within an Application Development Manager/400 session, you would specify SRVPGM instead of *SRVPGM, and PGM instead of *PGM.)

AppDict Services/400 also keeps statistics for each program you create such as:

- Creation date
- Source update date
- Program size and storage size
- Number of machine instructions (MI)
- Number of Object Definition Table (ODT) entries
- Program state and domain
- Sort sequence

To create an OPM program or ILE program using AppDict Services/400 directly or from within an Application Development Manager/400 session:

1. Start AppDict Services/400 in the dictionary which documents the library containing the source file for the program you want to create.
2. From the Application Dictionary Service/400 menu, type 3 (Work with programs and modules).
The Subset Programs and Modules to Work With display appears.
3. Specify the subsetting options needed, and press the Enter key to proceed to the Work with Programs and Modules display.
4. Press F6 (Create).

The Create Program, Service Program, or Module display appears.

Create Program, Service Program, or Module

Dictionary : ADSLIB04DT

Type choices, press Enter.

Object	<u>NEWMOD</u>	Name
Library	<u>ADSLIB05</u>	Name
Type	<u>*MODULE</u>	*PGM, *SRVPGM, *MODULE
Attribute	<u>CLLE</u>	Attribute, F4 for list
Source file	<u>QCLSRC</u>	Name
Library	<u>ADSLIB05</u>	Name
Source member	<u>NEWMOD</u>	Name
Text	<u>New module</u>	

F3=Exit F4=Prompt F5=Refresh F12=Cancel

Figure 17. Create Program, Service Program, or Module Display

Application Development Manager/400 Session

Create Program, Service Program, or Module

Dictionary : ADSLIB04DT

Project : ADMPRJ01

Group : MASTER

Type choices, press Enter.

Part	<u>NEWMOD</u>	Name
Type	<u>MODULE</u>	PGM, SRVPGM, MODULE
Attribute	<u>CLLE</u>	Attribute, F4 for list
Source part	<u>QCLSRC</u>	Name
Source file	<u>NEWMOD</u>	*TYPE, Name
Text	<u>New module</u>	

F3=Exit F4=Prompt F5=Refresh F12=Cancel

Figure 18. Create Program, Service Program, or Module Display in an Application Development Manager/400 Session



Tips:

1. If you are recreating a program, service program, or module that you previously created in Application Dictionary Services/400, the parameters that you used earlier, are used again.
2. If you are recreating a program, service program, or module that is documented in Application Dictionary Services/400 but was not previously created within AppDict Services/400, AppDict Services/400 retrieves the parameters from the documented program, service program, or module.

Exceptions

From the documented program, service program, or module AppDict Services/400 will not retrieve any special options that are not the defaults, such as:

- a. Entry module (ENTMOD(*FIRST) is the default)
- b. Binding directory (*BNDDIR)
- c. Export source file (EXPORT(*ALL) is the default)

However, if you prompt on the rebuild action and change the options, AppDict Services/400 stores the new build command and reuses it for future builds.

3. Remember the following when you are creating service programs, modules, or ILE SQL programs:
 - When you are creating new service programs in Application Development Manager/400 sessions, ensure that you have a valid export list (BNDSRC part) for that service program in your library list.
 - When creating new service programs or modules in Application Development Manager/400 sessions, ensure that you have a valid BLDOPt part for the source, to ensure Application Development Manager/400 builds the appropriate part.
 - If you are creating an ILE SQL program, you must ensure that there is a valid export list in the library list (in a source file named QSRVSRc), and with a source member having the same name as the service program to be created. (The CRTSQLxxxI commands do not prompt for an export list; it is assumed that the export list already exists.)

-
5. If you are working directly in AppDict Services/400, fill in the prompts as described in step 5a. If you are working with AppDict Services/400 in an Application Development Manager/400 session, fill in the prompts as described in step 5b on page 44.
 - a. If you are working directly in AppDict Services/400, fill in the prompts on the display as follows, and press the Enter key:
 - 1) In the *Object* prompt, type the name you want to assign to the program, service program, or module (NEWMOD, in this case).
 - 2) In the *Object Library* prompt, type the name of the library in which you want this program to be documented (ADSLIB05, in this case). The library must already be documented in the dictionary in which you are working.

3) In the *Type* prompt, specify the type of program (*MODULE, in this example). To create an OPM or ILE program, type *PGM. To create a service program or module, type *SRVPGM or *MODULE.

Note: If you specified *PGM as the type, the attribute of the first module you define will be used, even if that is not the attribute you specify in the *Attribute* prompt (next step).

4) In the *Attribute* prompt, type the attribute that describes the program's language, or press F4 for a list of attributes from which you can make a selection. (For example, type CLLE.)

5) In the *Source file*, *Library*, and *Source member* prompts, type the name of the source file, the library that contains the source file, and the name of the source member (QCLSRC, ADSLIB05, and NEWMOD, in this case).

Note: If this were a service program, you would type in the name of the source file that contains the export list for the service program. If this were an ILE SQL object (program, service program, or module), you would type the name of the member that contains the source with the embedded SQL statements.

6) In the *Text* prompt, type a description. (For this example, type New module.)

b. If you are working with AppDict Services/400 in an Application Development Manager/400 session, fill in the prompts on the display as follows, and press the Enter key:

1) In the *Part* prompt, type the name that you want to assign to the program, service program, or module in AppDict Services/400 (NEWMOD, in this case).

2) In the *Type* prompt, specify the type of program. To create an OPM or ILE program, type PGM. To create a service program or module, type SRVPGM or MODULE. (Type MODULE in this case.)

Note: If you specified PGM as the type, the attribute of the first module you define will be used even if that is not the attribute you specify in the *Attribute* prompt (next step).

3) In the *Attribute* prompt, type the attribute that describes the program's language, or press F4 for a list of attributes from which you can make a selection. (In this case, type CLLE.)

4) In the *Source part* and *Source file* prompts, type the name of the source part, and the name of the source file (QCLSRC and NEWMOD, in this case). If the source part does not already exist, AppDict Services/400 will create the part for you.

5) In the *Text* prompt, type a description. (For this example, type New module.)

The SEU Edit display appears.

6. If you want to edit your source file, you can do so at this time. Otherwise, press F3 to exit SEU.

The Exit display appears.

7. In the *Change/create member* prompt, type Y, and press the Enter key.

One of the following types of displays will appear:

- If you are creating an OPM program, the create program display appears for the language in which the program was written. For example, for RPG programs, the Create RPG/400 Program (CRTRPGPGM) display appears.
- If you are creating an ILE program, the AS/400 Create Program (CRTPGM) display appears.
- If you are creating a service program, the AS/400 Create Service Program (CRTSRVPGM) display appears.
- If you are creating a module, the AS/400 Create Module display appears for the particular type of module you are creating. For example, if you are creating an RPG module, the Create RPG Module (CRTRPGMOD) display appears.
- If you are creating an ILE SQL program, service program, or module, the appropriate display for the type of ILE SQL object you are creating appears. For example, if you are creating an ILE COBOL module with embedded SQL statements, the Create SQL ILE COBOL Object (CRTSQLCBLI) display appears.

8. Enter the required data in the Create Program (CRTPGM), Create Service Program (CRTSRVPGM), or create module display that appears.



Tips:

1. If you are creating an ILE program or service program and you did not remember to compile your modules first, you should stop creating the program at this point and recompile them now.
2. The CTRTBNDDxxx command is not supported in AppDict Services/400 because this command creates a temporary module that is deleted before the program is created. It is recommended that you do not document programs previously created with this command.
3. SQL objects (OPM programs, and ILE programs, service programs, and modules) must be created inside AppDict Services/400 for AppDict Services/400 to use the proper build command.

9. As well as defining all of the modules contained in the ILE program or service program, you also need to remember to define the following:

- If you are creating an ILE program, remember to define the program entry point (PEP) to the ILE program in the *Program entry procedure module* prompt.
- If you are creating a service program, remember to define the data items and procedures that you want to make available to other programs in the export prompts provided.

10. If you do not specify a source file in the Create Service Program (CRTSRVPGM) display:

- ALL* is used as the default parameter for the *Export* parameter. This means that all of the data items and procedures contained in the service program will be accessible to other programs.

- SEU is started for the specified source file
 - The service program is created with the specified source file as the export list. (In this case, you should specify the file that contains the binder language.)
- Note:** You do need an export list for SQL service programs, and since AppDict Services/400 does not prompt you to edit this, it must exist before you create the SQL service program.

11. Press the Enter key.

The Submit Job (SBMJOB) display appears.

12. Press the Enter key again.

A batch job is submitted to create the program object.

Creating Files

This section describes:

- The types of files you can create
- How to:
 - Create field reference files from templates
 - Create logical files
 - Specify the types of access for files
 - Create SQL files

Types of Files You Can Create

The different types of files that you can create in AppDict Services/400 are as follows:

File Attribute	File Type Created
DDMF	DDM File
DSPF	Display file
ICFF	Inter-System Communications Function File
LF	Logical file
LF-JOIN	Logical joined file
LF-SQLINX	Logical SQL index file
LF-SQLIVW	Logical SQL view file
PF-DTA	Physical data file
PF-FRF	Physical field reference file
PF-SQLTBL	Physical SQL table file
PF-SRC	Physical source file
PRTF	Printer file
SAVF	Save file

Creating a Field Reference File From a Template

Note: The information in this section does not apply to an Application Development Manager/400 session.

You can create a new physical file based on a field reference file. A **field reference file** can be used as the single location where you define all of the field definitions used in your programs, service programs, or modules. Field reference files are useful because:

- Instead of having to redefine the field each time you want to use it, you only need to refer to the field reference file
- If you want to change a field definition, you only need to change it in the field reference file

To make the creation of field reference files simpler, you can define a **field reference file** to be used as a template for creating other field reference files in your AS/400 system. AppDict Services/400 allows you to select which fields and key fields you want to include in field reference files that you create based on a template.

Note: You can use only the physical files with the attributes PF-FRF, PF-DTA, and PF-SQLTBL as the templates.

After you have defined a physical file, you can create logical files using the SEU editor.

In some cases, you will need to mark your field reference files so that AppDict Services/400 will recognize them as such. For more information about marking field reference files, refer to Chapter 6, “Marking Field Reference Files” on page 65.

This section describes how to:

- Create a physical file for a new field reference file
- Define file-level keywords
- Define record-level keywords
- Define fields
- Identify key fields
- Save the data description specifications (DDS) source

Creating a Physical File For a New Field Reference File

To create a field reference file from a template, you need to create a physical file with an attribute type of PF-DTA:

1. From the Work with Files display, press F6 (Create).

The Create File display appears.

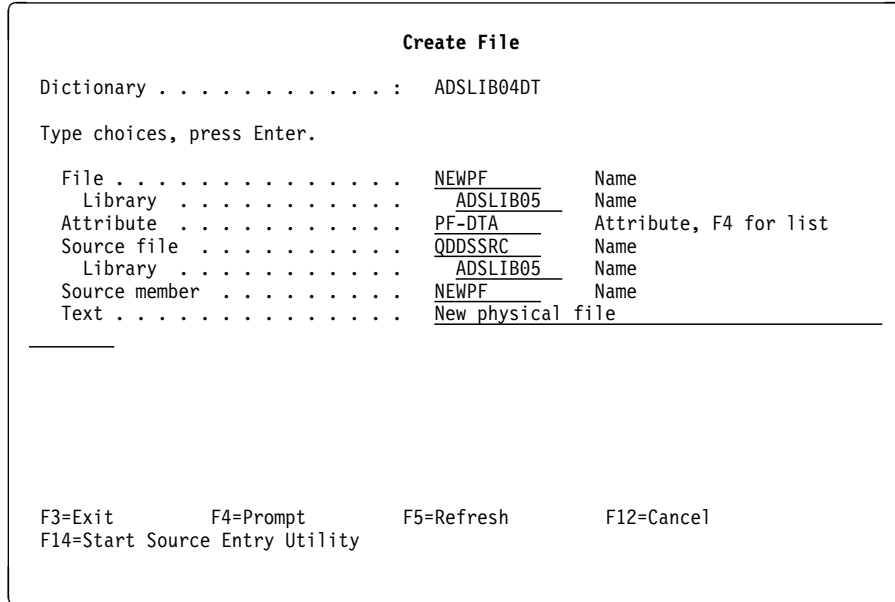


Figure 19. Create File Display. This example shows a physical file being created.

Note: The default value for the *Library* field in the Create File display is the value it had when you last used the display.

2. Fill in the prompts as follows, and press the Enter key:
 - a. In the *File* prompt, type the name of the file you want to create (NEWPF in this case).
 - b. In the *File Library* prompt, type ADSLIB05.
 - c. In the *Attribute* prompt, type PF-DTA.
 - d. In the *Source file* prompt, type the name of the source file containing the source member (QDDSSRC in this case).
 - e. In the *Source file Library* prompt, type ADSLIB05.
 - f. In the *Source member* prompt, type the new source member name NEWPF.
 - g. In the *Text* prompt, type New physical file.

The Design Physical File menu appears.

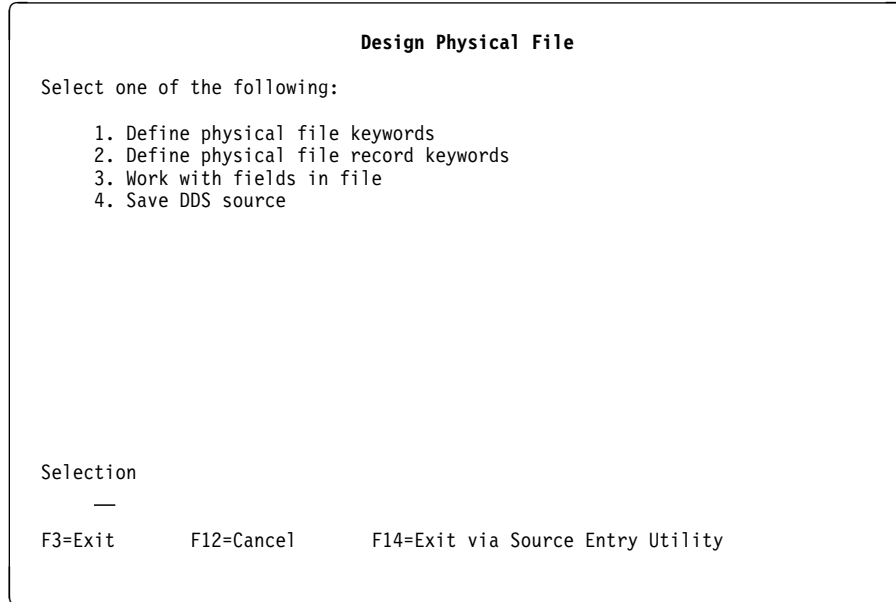


Figure 20. Design Physical File Menu

Defining File-Level Keywords

If you need to specify the file-level keywords for the physical file you want to create:

1. From the Design Physical File menu, select 1 (Define physical file keywords), and press the Enter key.

The Define Physical File Keywords display appears.

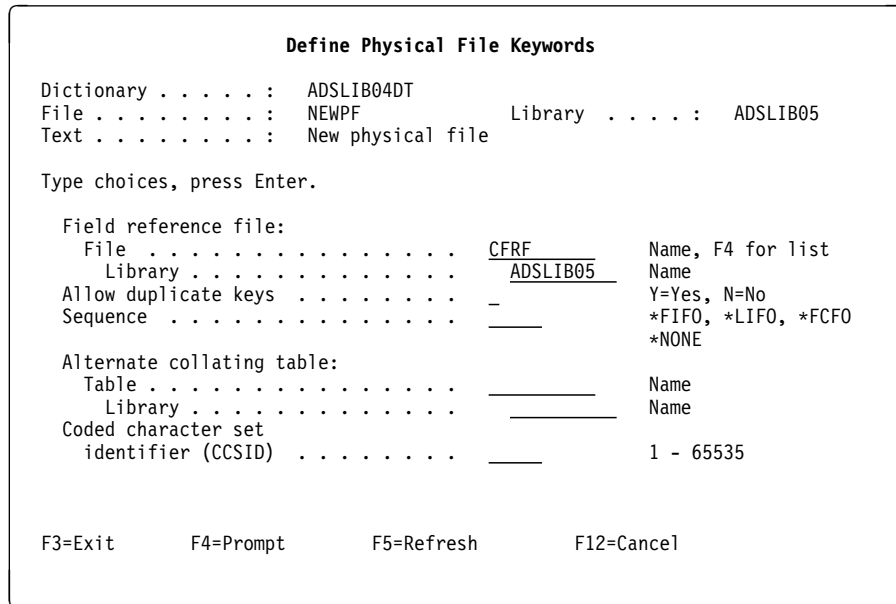


Figure 21. Define Physical File Keywords Display. This example shows physical file keywords being defined.

In this example, the physical file will be created based on a field reference file, so type the names of the file and the library where the field reference file is located (CFRF in ADSLIB05 in this case), and press the Enter key.

You are returned to the Design Physical File menu.

Defining Record-Level Keywords

After you have specified the file-level keywords for your physical file, you can define its record-level keywords:

1. From the Design Physical File menu, type 2 (Define physical file record keywords), and press the Enter key.

The Define Physical File Record Keywords display appears.

Define Physical File Record Keywords

Dictionary : ADSLIB04DT
File : NEWPF Library : ADSLIB05
Text : New physical file

Type choices, press Enter.

Record : NEWRCD Name
Text : New record format name

Use record format in:
File : _____ Name, F4 for list
Library : _____ Name

F3=Exit F4=Prompt F5=Refresh F12=Cancel

Figure 22. Define Physical File Record Keywords Display. This example shows record-level keywords being defined.

2. Specify the new record name (NEWRCD in this case), and the text (New record format name in this case), for the physical file you want to create and press the Enter key.

You are returned to the Design Physical File menu.

If you want to refer to the record format of a physical file that already has been defined, specify the name of the physical file in the *File* field and the name of the library in the *Library* prompt.

Defining Fields

After you have specified the record-level keywords for the physical file, you can define its fields:

1. From the Design Physical File menu, type 3 (Work with fields in file), and press the Enter key.

The Work with Fields in File display appears. This display is empty, since there are currently no fields defined for this file.

```

                                Work with Fields in File
Dictionary . . . . . : ADSLIB04DT
File . . . . . : NEWPF          Library . . . . . : ADSLIB05
Text . . . . . : New physical file
Record format . . . . . : NEWRC

Type options or orders, press Enter.
  2=Change          3=Copy          4=Delete
  6=Mark as key field  8=Define keywords

Opt Order   Field      Reference Type Length Decimal Key sequence
(No fields in file)

F3=Exit          F5=Refresh    F6=Create      F10=Command entry
F11=Display text F12=Cancel    F24=More keys

```

Figure 23. Work with Fields in File Display

2. There are two ways you can specify fields:

- Use F6 (Create) to create field definitions to be added to this record format.
- Use F13 (Add fields from another file) to select existing field definitions from another file.

In this example we will select fields from another file.

3. Press F13 (Add fields from another file).

The Select Files to Add Fields display appears.

```

                                Select Files to Add Fields
Dictionary . . . . . : ADSLIB04DT
File . . . . . : NEWPF          Library . . . . . : ADSLIB05
Text . . . . . : New physical file
Position to . . . . . : _____ Starting characters of file

Type options, press Enter.
  1=Select

Opt File      Library   Attribute  Text
  1 CFRF       ADSLIB05  PF-FRF     CUSTOMER DATABASE FIELD REFERENCE FI
  - CPF        ADSLIB05  PF-DTA     CUSTOMER DATABASE FILE.
  - CUSFRF     ADSLIB05  PF-FRF     CUSTOMER DATABASE FIELD REFERENCE FI
  - CUSMPF     ADSLIB05  PF-DTA     CUSTOMER DATABASE FILE.
  - EXFILE     ADSLIB05  PF-FRF     CUSTOMER DATABASE FILE.
  - OLDPF     ADSLIB05  PF-FRF     NEW PHYSICAL FILE

F3=Exit          F5=Refresh    F12=Cancel    F17=Subset
Bottom

```

Figure 24. Select Files to Add Fields Display

4. Type 1 (Select) beside the names of any files you want to use as a basis for defining the fields on the new file (CFRF in this case), and press the Enter key.

The Select Fields from Files display appears, listing the fields that you can add to the new file.

Select Fields from Files

Dictionary : ADSLIB04DT
 File : CFRF Library : ADSLIB05
 Text : CUSTOMER DATABASE FIELD REFERENCE FILE

Type options, press Enter.
 1=Select

Opt	Field	Type	Length	Decimal	Text
1	ADDR	A	20		CUSTOMER ADDRESS
-	ARBAL	P	10	2	ACC. RECEIVABLE BALANCE
1	CITY	A	20		CUSTOMER CITY
1	CUST	A	5		CUSTOMER NUMBER
-	CUSTYP	A	1		CUSTOMER TYPE
1	NAME	A	20		CUSTOMER NAME
-	SRHCOD	A	3		CUSTOMER NAME SEARCH CODE
1	STATE	A	2		CUSTOMER STATE
1	ZIPCODE	P	5	0	CUSTOMER ZIP CODE
-	AGE	P	3		AGE

Bottom

F3=Exit F5=Refresh F12=Cancel F13=Repeat

Figure 25. Select Fields from Files Display. This example shows fields being selected to be included in the new file.

5. Type 1 (Select) beside the fields that you want to include in the new file you are creating (ADDR, CITY, CUST, NAME, STATE, and ZIPCODE in this case), and press the Enter key.

The Work with Fields in File display appears, showing the newly selected fields.

Identifying Key Fields

Once you have defined the fields in the new file, you can identify those that you want to mark as key fields:

```

Work with Fields in File

Dictionary . . . . . : ADSLIB04DT
File . . . . . : NEWPF          Library . . . . . : ADSLIB05
Text . . . . . : New physical file
Record format . . . . . : NEWRCD

Type options or orders, press Enter.
  2=Change          3=Copy          4=Delete
  6=Mark as key field  8=Define keywords

Opt Order   Field      Reference Type Length Decimal Key sequence
-  -  -  -  -  -  -  -  -  -  -  -  -
  6  10  ADDR         R      A   20
  6  20  CITY         R      A   20
  6  30  CUST         R      A    5
  6  40  NAME         R      A   20
  6  50  STATE        R      A    2
  6  60  ZIPCODE      R      P    5    0

Bottom

F3=Exit          F5=Refresh      F6=Create       F10=Command entry
F11=Display text F12=Cancel      F24=More keys
  
```

Figure 26. Work with Fields in File Display. This example shows key fields being selected for definition.

1. From the Work with Fields in File display, type 6 (Mark as key field) beside the fields that you want to mark as key fields (CITY, CUST, STATE and ZIPCODE in this case), and press the Enter key.

The Work with Key Fields display appears. The key sequence appears in the *Sequence* column.

```

Work with Key Fields

Dictionary . . . . . : ADSLIB04DT
File . . . . . : NEWPF          Library . . . . . : ADSLIB05
Text . . . . . : New physical file

Type options or sequences, press Enter.
  2=Change keywords  4=Drop

Opt Sequence Field      Descend Absolute Signed Digit
-  -  -  -  -  -  -  -  -  -  -  -
  6  5  CITY         -
  6 10  CUST         -
  6 15  STATE        -
  6 20  ZIPCODE      -

Bottom

F3=Exit          F5=Refresh      F10=Command entry
F11=Display keywords F12=Cancel      F24=More keys
  
```

Figure 27. Work with Key Fields Display. This example shows key fields being selected to change keywords.

2. To change the key field sequence or the key field keywords, press F14 (Work with key fields) on this display.
3. To change key field keywords, type 2 (Change keywords) beside the fields that you chose as key fields (ZIPCODE in this case), and press the Enter key.

The Change Key Field Keywords display appears.

On this display, you can also change the sequence of the key fields by re-ordering the sequence numbers in the **Sequence** column. If you change the sequence and type an option at the same time, the sequence will be changed after the option runs.

Change Key Field Keywords

Dictionary : ADSLIB04DT
 Field : ZIPCODE Type : P
 File : NEWPF Library : ADSLIB05
 Text : New physical file

Type choices, press Enter.

Descending sequence	Y	Y=Yes, N=No
Absolute value	N	Y=Yes, N=No
Signed	N	Y=Yes, N=No
Digit	N	Y=Yes, N=No
Zoned	N	Y=Yes, N=No
No alter sequence	N	Y=Yes, N=No
Unsigned	N	Y=Yes, N=No

F3=Exit F5=Refresh F12=Cancel

Figure 28. Change Key Field Keywords Display. This example shows key field keywords being changed.

In this example you will specify key field keywords by the requirements you have for this physical file.

4. In the *Descending sequence* prompt, type Y to specify that the values of this key field (ZIPCODE in this case) will be obtained in descending order, and press the Enter key. For information about the key field keywords, refer to the online help.

You are returned to the Work with Key Fields display.

5. Press F3 (Exit) to go to the Work with Fields in File display.
6. Press the Enter key.

The Design Physical File menu appears.

Saving the Data Description Specifications Source

You can save the generated data description specifications (DDS) in a source member.

Note: If you do not save the DDS, the previous file creation or change will be canceled.

1. From the Design Physical File menu, type 4 (Save DDS source), and press the Enter key.

If you prefer to exit from the menu now, either press F3 (Exit) if you do not want to create the file, or F14 (Exit via Source Entry Utility) to call the SEU editor, and exit from there.

Option 40 (Display generated DDS) appears on your screen.

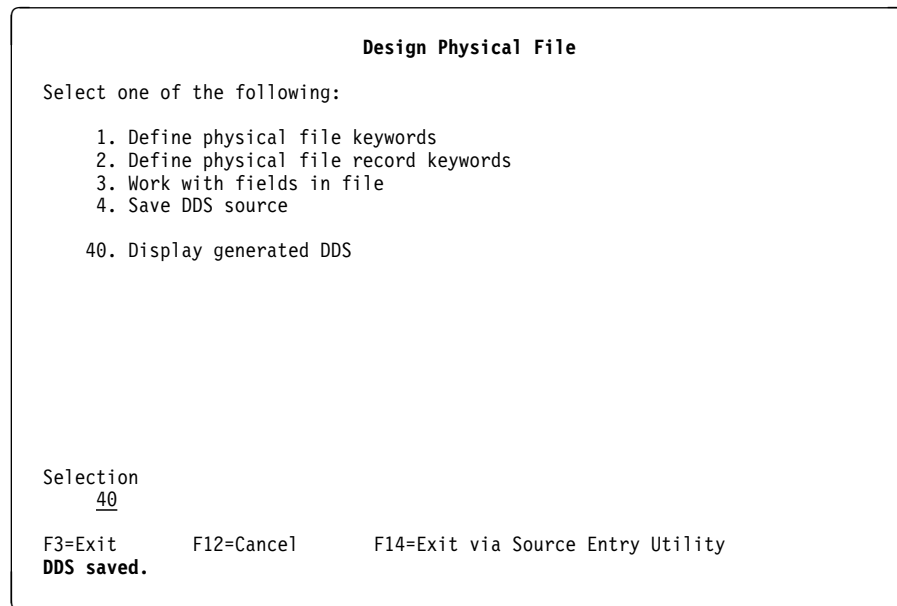


Figure 29. Design Physical File Menu

2. Type option 40 (Display generated DDS) to verify the source, and press the Enter key.

The Display Generated DDS display appears, where you can review the DDS source.


```

                                Display Generated DDS
Dictionary . . . . . : ADSLIB04DT
File . . . . . : NEWPF
Library . . . . . : ADSLIB05      Attribute . . . . . : PF-DTA
Text . . . . . : New physical file

*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...
A* New physical file
A* DSNDBF 10:25:15                CPF= R04M00 900928
A                                REF(ADSLIB05/CFRF)
A                                TEXT('New record format name')
A      R NEWRCD                   TEXT('CUSTOMER ADDRESS')
A      ADDR      R                TEXT('CUSTOMER CITY')
A      CITY      R                TEXT('CUSTOMER NUMBER')
A      CUST      R                TEXT('CUSTOMER NAME')
A      NAME      R                TEXT('CUSTOMER STATE')
A      STATE     R                TEXT('CUSTOMER ZIP CODE')
A      ZIPCODE   R
A      K CITY

Press Enter to continue.
F3=Exit      F6=Print      F12=Cancel

More...

```

Figure 30. Display Generated DDS Display. This example shows the DDS source being verified.

After you have reviewed the DDS source, you can change it.

3. Press F3 (Exit) to return to the Design Physical File menu.
4. Press F14 (Exit via Source Entry Utility) to go to the SEU Edit display.
5. Change the field ZIPCODE to ZIP.
6. After exiting the SEU editor, the Create Physical File (CRTPF) display appears, where you can change any of the CRTPF parameters according to your requirements. The AppDict Services/400 feature will store the create command string in the dictionary and will use it to recreate the physical file next time.
7. Press the Enter key.

The Submit Job (SBMJOB) display appears.
8. To submit a batch job to compile the physical file, press the Enter key again.

The Work with Files display appears. You need to wait for a while for the dictionary to be synchronized.
9. Press F5 (Refresh) to update the list of files on the Work with Files display.
10. In the *Position to* prompt, type NEWPF, and press the Enter key.

The new physical file, NEWPF, is listed on the top of the Work with Files display.

Creating a Logical File

You can also create logical files by pressing F6 (Create) from the Work with Files display. Fill in the prompts described in “Creating a Physical File For a New Field Reference File” on page 47, except type LF in the *Attribute* prompt, instead of PF-DTA. This function will call the SEU editor, where you can specify the DDS source needed to define the logical file. The AppDict Services/400 feature does not provide specific support to create logical file definitions.

Specifying the Types of Access for Files

As you develop your programs, you might need to change the type of access for the files that they use. The types of access you can choose from are:

- Any type of access
- Input
- Output
- Update
- Unspecified
- External data structure

To change the access type of a file:

1. From the Application Dictionary Services/400 menu, type 2 (Work with files), and press the Enter key.

The Subset Files to Work With display appears.

2. Specify the subsetting criteria that you want, and press the Enter key.

The Work with Files display appears.

3. Beside the file for which you want to change the type of access, type 10 (Work with programs referencing file), and press the Enter key.

The Work with Programs Referencing File display appears. The file with which you are working is shown at the top of the display.

4. Press F19 (Select file usage).

The Specify Types of Access for File display appears.

5. Type Y beside an access type to select it, or N to deselect it, and press the Enter key.

The access types you selected are activated for the file, and the Work with Programs Referencing File display appears again.

Creating SQL Files

You can build SQL table files based on a field reference file, and then create SQL index files and SQL view files based on the SQL table files and physical files.

Note: The information in this section does not apply to an Application Development Manager/400 session because the Application Development Manager/400 feature does not support SQL files.

This section shows you, through examples, how to:

- Create an SQL table file based on a field reference file
- Create an SQL index file based on an SQL table file
- Create an SQL index file based on a physical file

Creating an SQL Table File Based on a Field Reference File

To create an SQL table file based on a field reference file and to select the fields defined in this field reference file:

1. From the Application Dictionary Services/400 menu, type 2 (Work with files) to get to the Subset Files to Work With display.
2. Specify the subsetting options needed, and press the Enter key to proceed to the Work with Files display.

3. Press F6 (Create) to get to the Create File display.
4. Type the name of the SQL table file you want to create (NEWSQLT in this case) and its attribute (PF-SQLTBL in this case), and press the Enter key.

The Work with SQL Table Fields display appears.

```

                                Work with SQL Table Fields
Dictionary . . . . . : ADSLIB04DT
File . . . . . : NEWSQLT           Library . . . . . : ADSLIB05
Text . . . . . : New SQL table file

Type options or orders, press Enter.
 2=Change           3=Copy           4=Delete
 8=Define keywords

Opt Order   Field      Type Length Decimal Text

(No fields in this SQL table file)

F3=Exit           F5=Refresh       F6=Create         F10=Command entry
F11=Display text  F12=Cancel       F24=More keys

```

Figure 31. Work with SQL Table Fields Display

This field list is empty, since you have not yet specified any fields to be used in this SQL table file. There are two ways you can create fields:

- Use F6 (Create) to create individual fields
- Use F13 (Add fields from another file) to select already defined fields from other files

In this example we will add fields from another file.

5. Press F13 (Add fields from another file).
The Select Files to Add Fields display appears. Refer to Figure 24 on page 51.
6. Type 1 (Select) beside the names of any files you want to use as a basis for defining the fields in the new file (CFRF in this case), and press the Enter key after you have selected all of the files you want.
The Select Fields from Files display will list the fields that you can add to the new file.
7. Type 1 (Select) beside the fields that you want to include in the SQL table files you are creating, and press the Enter key.

Note: If you want to select all the fields, type 1 (Select) in the first option field, and then press F13 (Repeat). All the option fields will be filled in with the number 1.

The Work with SQL Table Fields display reappears. This display now lists the fields you selected.

```

Work with SQL Table Fields
Dictionary . . . . . : ADSLIB04DT
File . . . . . : NEWSQLT Library . . . . . : ADSLIB05
Text . . . . . : New SQL table file

Type options or orders, press Enter.
  2=Change          3=Copy          4=Delete
  8=Define Keywords

Opt  Order  Field      Type  Length  Decimal  Text
--  -
  10  ADDR    A        20
  20  ARBAL   P        10    2  ACC. RECEIVABLE BALANCE
  30  CITY    A        20  CUSTOMER CITY
  40  CUST    A         5  CUSTOMER NUMBER
  50  CUSTYP  A         1  CUSTOMER TYPE
  60  NAME    A        20  CUSTOMER NAME
  70  SRHCO   A         3  CUSTOMER NAME SEARCH CO
  80  STATE   A         2  CUSTOMER STATE
  90  ZIPCO   P         5    0  CUSTOMER ZIP CODE

F3=Exit          F5=Refresh      F6=Create      F10=Command entry
F11=Display text F12=Cancel      F24=More keys

Bottom

```

Figure 32. Work with SQL Table Fields Display. This example shows the selected fields.

Note: If you want to change the sequence in the *Order* prompt to reorder these fields, just type in the numbers that represent the new sequence that you want.

8. To go to the DB2/400 Query Manager menu, press the Enter key.

Note: If you press F3 (Exit), you are returned to the Work with Files display; if you press F12 (Cancel), the Create File display appears.

9. To go to the Submit Job (SBMJOB) display, press the Enter key.

10. To submit a batch job to create the SQL table file NEWSQLT, press the Enter key.

When the job is complete, the Work with Files display will list the new file.

Creating an SQL Index or View File Based on an SQL Table File

You can create an SQL index or SQL view file based on an SQL table file.

Notes:

1. You can only use physical files with the attribute PF-DTA or PF-SQLTBL as the templates to create your SQL index files.
2. You can also follow the procedure above to create an SQL view file based on a physical file.

To create an SQL index or SQL view file based on an SQL table file.

1. From the Work with Files display, press F6 (Create) to go to the Create File display.
2. Type the name of the SQL index file that you want to create (NEWSQLINX in this case) and its attribute (LF-SQLINX in this case), and press the Enter key.

The Define SQL Index Keywords display appears.

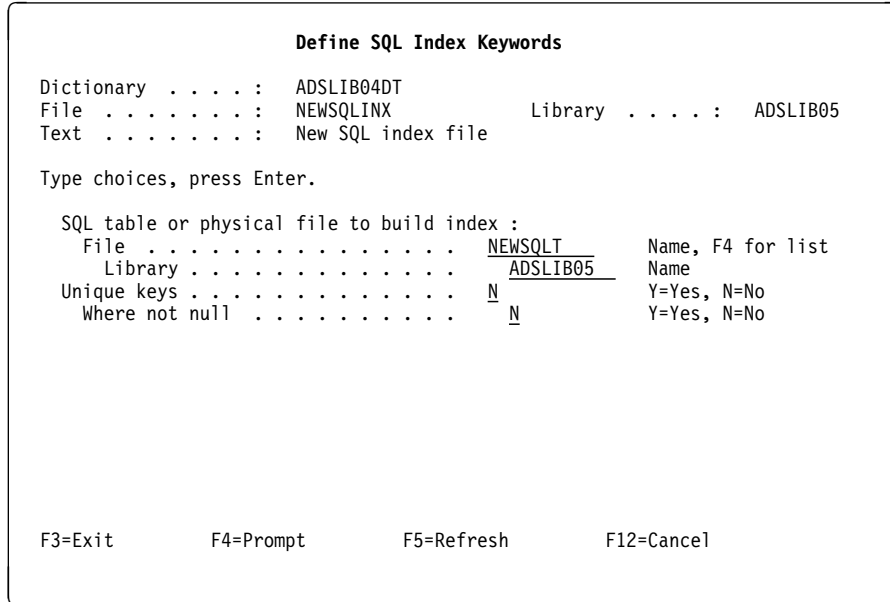


Figure 33. Define SQL Index Keywords Display

3. Fill in the prompts as follows, and press the Enter key:
 - a. In the *File* prompt, type the name of an SQL table file you have previously defined (NEWSQLT in this case).
 - b. In the *Library* prompt, type ADSLIB05.

The Work with SQL Index Fields display will list the fields in NEWSQLT.

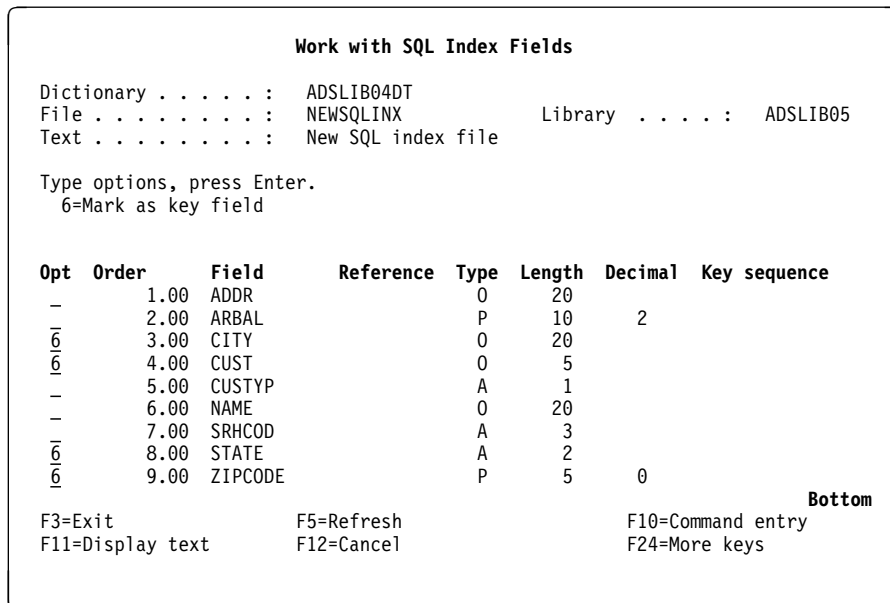


Figure 34. Work with SQL Index Fields Display. This example shows key fields being selected.

4. Type 6 (Mark as key field) beside the fields that you want to mark, and press the Enter key.

The sequence numbers of the key fields appear in the *Key sequence* prompt.

5. To go to the DB2/400 Query Manager menu, press the Enter key.
6. To go to the Submit Job (SBMJOB) display, press the Enter key again.
7. To submit a batch job to create the SQL index file NEWSQLINX, press the Enter key.

When the job is completed, the Work with Files display will list the new file.

Creating an SQL Index or View File Based on a Physical File

You can create an SQL index or view file based on a physical file. The steps are the same as those described for an SQL table file above, except that you specify the name of a physical file instead of an SQL table file in the *File* prompt on the Define SQL Index Keywords display.

Creating Other Types of Objects

In AppDict Services/400, you can also create objects other than programs, files, and fields. The objects you can create include:

- Binding directories
- Data, message, and output queues
- Classes
- Commands
- Journals and journal receivers
- Menus
- Data areas
- QUERY Manager forms and queues
- Panel groups
- Search indexes
- Tables

To create such objects:

1. From the Application Dictionary Services/400 menu, type 4 (Work with other objects), and press the Enter key.

The Subset Objects to Work With display appears.

2. Specify the subsetting criteria that you want, and press the Enter key.

The Work With Other Objects display appears.

3. Press F6 (Create).

The Create Object display appears.

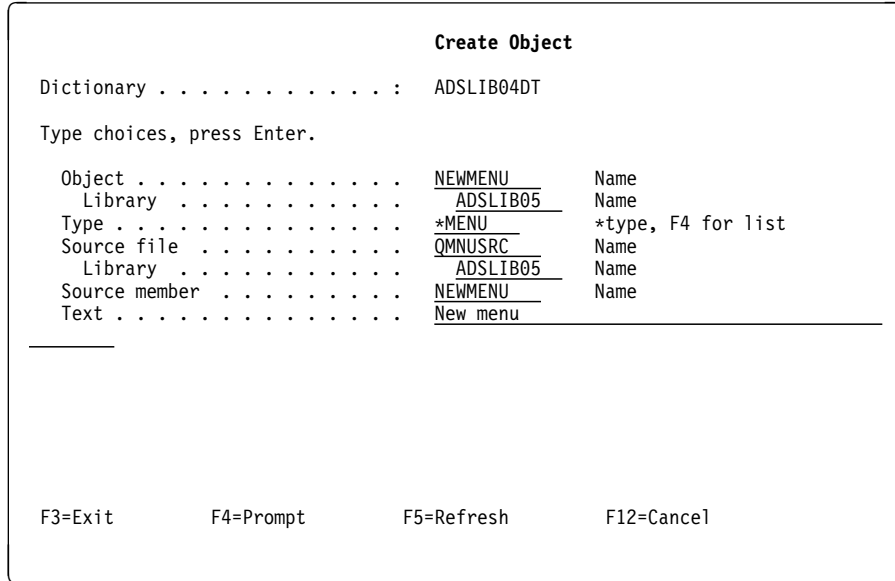


Figure 35. Create Object Display

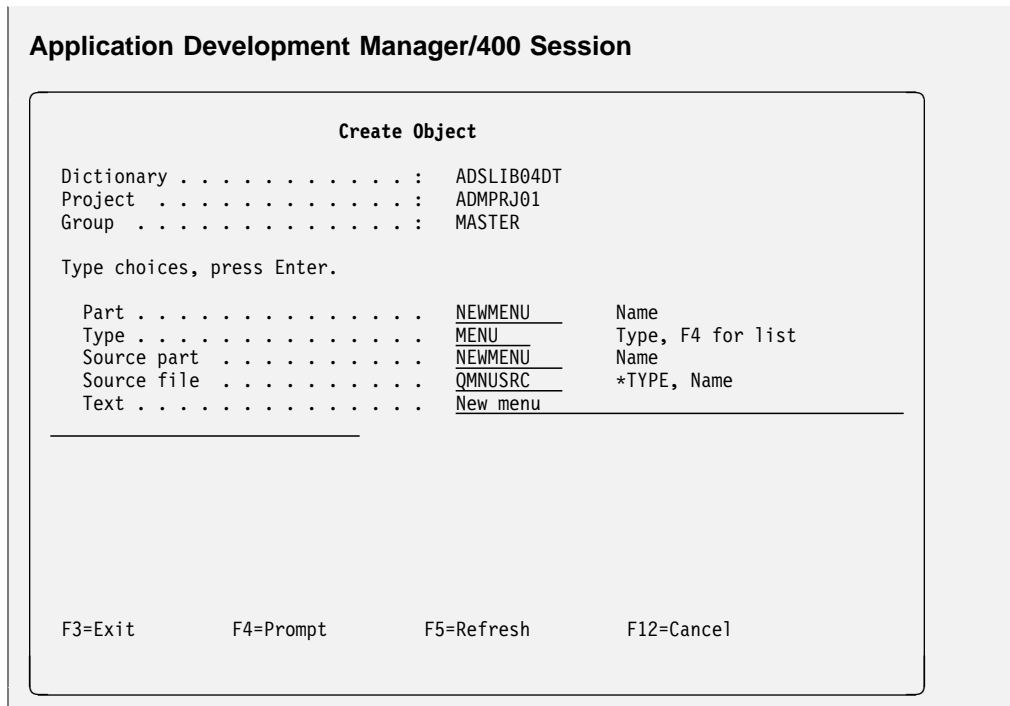


Figure 36. Create Object Display in an Application Development Manager/400 Session

4. If you are working directly in AppDict Services/400, fill in the prompts as described in step 4a. If you are working with AppDict Services/400 in an Application Development Manager/400 session, fill in the prompts as described in step 4b on page 63.
 - a. If you are working directly in AppDict Services/400, fill in the prompts as follows, and press the Enter key:
 - 1) In the *Object* prompt, type the name you want to give to the new object in AppDict Services/400 (NEWMENU, in this case).

- 2) In the *Object Library* prompt, type the name of the library (ADSLIB05, in this case) in which you want this object documented. The library must already be documented in the dictionary in which you are working.
- 3) In the *Type* prompt, specify the type of object that you want to create, or press F4 for a list of types from which you can select. (For this example, type *MENU.)



Tip:

You do not need to specify a source file for most of the types of objects that you can create. If you do specify a source file for these types of objects, it is ignored. However, to create some objects, such as menus, you do need to specify a source file.

- 4) In the *Source file*, *Library*, and *Source member* prompts, type the name of the source file, the library that contains the source file, and the name of the source member (QMNUSRC, ADSLIB05, and NEWMENU, in this case).
 - 5) In the *Text* prompt, type a brief description of the object. (Type New menu for this example.)
- b. If you are working in AppDict Services/400 from within an Application Development Manager/400 session, fill in the prompts as follows, and press the Enter key.
- 1) In the *Part* prompt, type the name you want to give to the new object in AppDict Services/400 (NEWMENU, in this case).
 - 2) In the *Type* prompt, specify the type of object you want to create, or press F4 for a list of types from which you can make a selection. (Specify MENU, in this case.)
 - 3) In the *Source part* and *Source file* prompts, type the name of the source part and source file for the object you are creating (NEWMENU and QMNUSRC, in this case).
 - 4) In the *Text* prompt, type a short description of the new object. (For this example, type New menu.)

The AS/400 displays that follow will vary depending on which object you are creating.

- c. Work through the AS/400 displays that appear.

After you have finished working through all of the necessary displays, a message appears indicating that the object has been created.

Compiling and Building Program Objects

This section describes how you can compile and build objects within AppDict Services/400.

Compiling Objects

You can use option 14 (Compile) from any of the many AppDict Services/400 displays on which it appears to compile modules, create (bind) ILE programs and service programs, and compile OPM programs. The 14 (Compile) option works similarly to the F6 (Create) function described in “Creating Programs, Service Programs, and Modules in Application Dictionary Services/400” on page 40.

The 14 (Compile) option works as follows:

- If you are recreating a program, service program, or module that you previously created in Application Dictionary Services/400, the parameters that you used earlier, are used again.
- If you are recreating a program, service program, or module that is documented in Application Dictionary Services/400 but was not previously created within Application Dictionary Services/400, Application Dictionary Services/400 retrieves the parameters from the documented program, service program, or module.

Exceptions

From the documented program, service program, or module AppDict Services/400 will not retrieve any special options that are not the defaults, such as:

1. Entry module (ENTMOD(*FIRST) is the default)
2. Binding directory (*BNDDIR)
3. Export source file (EXPORT(*ALL) is the default)
4. If you are creating an ILE SQL program, you must ensure that there is a valid export list in the library list (in a source file named QSRVSRG), and with a source member having the same name as the service program to be created. (The CRTSQLxxx commands do not prompt for an export list; it is assumed that the export list already exists.)

However, if you prompt on the rebuild action and change the options, AppDict Services/400 stores the new build command and reuses it for future builds.

Building Objects

If you are working with AppDict Services/400 from within an Application Development Manager/400 session, you can use option 14 (Build) that appears on several of the displays. This option builds one or more parts in batch using the BLDPART command with the SCOPE(*NORMAL) parameter. When a part is built using this option, the information about the associated source part is passed to the BLDPART command. You must have *UPDATE access to the working group. The built parts are placed in the working group.

Chapter 6. Marking Field Reference Files

In AppDict Services/400, you can use field reference files to define your program fields, and then refer to the field reference file from your programs, instead of redefining your fields each time they occur.

You also need to mark field reference files (FRF) so that AppDict Services/400 will recognize them as such. Doing so does not change their definition, except that AppDict Services/400:

- Changes the record capacity of the file to 1
- Removes all of the members
- Changes the attribute of the file to PF-FRF in the AppDict Services/400 environment



Tips:

1. When you create a physical file with no member, AppDict Services/400 automatically marks it as a field reference file.
 2. Marked files contain no records and have no logical file dependencies.
-

Marking New Field Reference Files

To mark a physical file as a field reference file:

1. Type `WRKAPPDCT ADM(*NO)`, or if you want to use AppDict Services/400 from within an Application Development Manager/400 session, type `WRKAPPDCT ADM(*YES)`, and press the Enter key.

The Work with Dictionaries display appears.

2. Type 10 (Mark field reference files) beside the dictionary with which you want to work (ADSLIB01DT in this case), and press the Enter key.
 - a. If you are in an Application Development Manager/400 session, press the Enter key to proceed to the Specify Project to Work With display.

Application Development Manager/400 Session

```

                Specify Project to Work With

Dictionary . . . . . : ADSLIB01DT

Type choices, press Enter.

Project . . . . . ADMPRJ01
Group . . . . . MASTER
Search path . . . . . *DFT      Name, *DFT

F3=Exit      F5=Refresh      F12=Cancel
    
```

Figure 37. Specify Project to Work With Display in an Application Development Manager/400 Session

- b. Specify the project, group, and search path in the *Project*, *Group*, and *Search path* fields respectively (in this case, ADMPRJ01, MASTER, and *DFT), and press the Enter key.

The Mark Field Reference Files display appears, showing a list of physical files documented in the dictionary that can be marked as field reference files. This display will list only those documented physical files that contain logical file dependencies.

```

                Mark Field Reference Files

Dictionary . . . . . : ADSLIB01DT

Type options, press Enter.
 1=Select

Opt File      Library      Text
 1 CUSFX      AP01.MST  Customer Master File
 - EMPPAY     AP01.MST  EMPLOYEE SALARY DATA FILE
 - EMPREF     AP01.MST  EMPLOYEE DATA FIELD REFERENCE FILE
 - JOE        AP01.MST  DBCS Field Test
 - JOHN       AP01.MST  DBCS Field Test
 - PDFFF1     AP01.MST  PRODUCT MASTER FILE
 - REFF       AP01.MST  Sales System FIELD REFERENCE FILE
 - SSTPF     AP01.MST  Data Type check

F3=Exit      F5=Refresh      F12=Cancel      F13=Repeat      Bottom
    
```

Figure 38. Mark Field Reference Files Display. This example shows files being selected.

```

Application Development Manager/400 Session

                                Mark Field Reference Files

Dictionary . . . . . : ADSLIB01DT
Project . . . . . : ADMPRJ01

Type options, press Enter.
  1=Select

Opt Part          Group          Text
  1  CUSFX         MASTER
  -  EMPPAY        MASTER
  -  EMPREF        MASTER
  -  JOE           MASTER
  -  JOHN          MASTER
  -  PDFFF1        MASTER
  -  REFF          MASTER
  -  SSTPF         MASTER

F3=Exit      F5=Refresh      F11=Display text
F12=Cancel   F13=Repeat

Bottom

```

Figure 39. Mark Field Reference Files Display in an Application Development Manager/400 Session. This example shows files being selected.

3. Type 1 (Select) beside the file you want to mark as a field reference file (CUSFX in this case), and press the Enter key.



Tip:

For an Application Development Manager/400 session, if the BLDOPT part for the source part does not exist, it will be created automatically; if it does exist, its contents will be edited. Another way to create a field reference file is:

1. Press F6 (Create) on the Work with Files display to get to the Create File display.
2. Type PF-FRF in the *Attribute* prompt, and press the Enter key.

You should change the BLDOPT part to add the option MBR(*NONE) for the Create Physical File (CRTPF) command. Otherwise, the file attribute will become PF-DTA when you recompile it next time.

Unmarking Previously Marked Field Reference Files

Sometimes you may want to remove the mark from a previously marked field reference file.

To do this, do the following:

- If you are working directly in an AppDict Services/400 session, delete the object and the object information from the dictionary, and then recreate the file with the MBR(*FILE) parameter.

The file will automatically be documented.

- If you are working in AppDict Services/400 from within an Application Development Manager/400 session, use option 14 (Build) to rebuild the part in AppDict Services/400. If there is a build option part associated with the file part, use the MBR(*FILE) parameter on the CRTLF command when rebuilding the part.

The file will automatically be documented.

Chapter 7. Managing the Impact of Changes to Fields

Once you have created a dictionary, you can use it to get detailed information about specific fields and about programs and files that would be affected by a change to a field. Conversely, if you make changes to fields or recreate any impacted files or programs, these changes will be reflected in the dictionary.

This chapter describes how to:

- Determine which *fields* would be affected by a change to another field by looking at the field reference hierarchy
- Determine which *files* would be affected by a change to a field
- Determine, for any given file that is affected by a field change, which programs contain references to that file
- Change a field
- Recreate impacted files, programs, service programs, and modules

Determining the Field Reference Hierarchy

The Application Dictionary Services/400 (AppDict Services/400) feature provides the capability to determine the **field reference hierarchy** of a field by identifying all other fields that refer to it, or are referred to by it, or both. Looking at the field reference hierarchy can be a good way of analyzing the effect that changing a field will have on other fields in program objects.

To determine all relevant field references, first select a field that has been documented in a dictionary, as follows:

1. From the Application Dictionary Services/400 menu, type 1 (Work with fields), and press the Enter key.

The Subset Fields to Work With display appears.

```

Subset Fields to Work With
Dictionary . . . . . : ADSLIB01DT
Type choices, press Enter.
Field . . . . . QTY          *ALL, name, generic*
Record . . . . . *ALL        *ALL, name, generic*
File . . . . . PDF           *ALL, name, generic*
Library . . . . . *ALL        *ALL, name, generic*
Attribute . . . . . *ALL       *ALL, attribute
                                           F4 for list
Text:
Search words . . . . . _____
                               _____
Search condition . . . . . 1      1=Or, 2=And

F3=Exit      F4=Prompt      F5=Refresh      F12=Cancel

```

Figure 40. Subset Fields to Work With Display

```

Application Development Manager/400 Session
Subset Fields to Work With
Dictionary . . . . . : ADSLIB01DT
Project . . . . . : ADMPRJ01
Type choices, press Enter.
Field . . . . . QTY          *ALL, name, generic*
Record . . . . . *ALL        *ALL, name, generic*
Part . . . . . PDF           *ALL, name, generic*
Attribute . . . . . *ALL       *ALL, attribute
                                           F4 for list
Text:
Search words . . . . . _____
                               _____
Search condition . . . . . 1      1=Or, 2=And

F3=Exit      F4=Prompt      F5=Refresh      F12=Cancel

```

Figure 41. Subset Fields to Work With Display in an Application Development Manager/400 Session

2. Fill in the prompts as follows, and press the Enter key:
 - a. In the *Field* prompt, type the name of the field you want (QTY, in this example).
 - b. In the *File* prompt, type the name of the file (PDF, in this case). (If you were working in AppDict Services/400 from within an Application Development Manager/400 session, you would type PDF in the *Part* prompt.)

c. In the other prompts (except the *Text* prompts), type *ALL.

Typing *ALL in the other prompts displays all of the fields documented in the dictionary. Specifying generic names in the rest of the prompts displays a subset of the fields in the dictionary. See the online help for information about specifying subsetting criteria.

Note: Search word prompts on all the displays are case-insensitive. Thus, for example, if you type NEWYORK in the *Search words* prompt, the strings newyork, newYORK, and NEWyork will also be found.

The Work with Fields display appears, listing the fields you requested.

```

                                Work with Fields
Dictionary . . . . . : ADLIB01DT
Position to . . . . . : _____ Starting characters of field

Type options, press Enter.
  2=Edit           5=Display           8=Work with impacted files
  9=Work with related files       12=Work with record formats ...

Opt Field      Record    File      Library    Attribute
19  QTY          PDFREC     PDF         AP01.MST    PF-DTA

Command _____ Bottom
====>
F3=Exit          F4=Prompt     F5=Refresh    F6=Add fields
F9=Retrieve      F12=Cancel    F23=More options F24=More keys
This is a subsetted list.

```

Figure 42. Work with Fields Display. This example shows the fields that match the criteria specified.

Application Development Manager/400 Session

```

                                Work with Fields
Dictionary . . . . . : ADSLIB01DT
Position to . . . . . : _____ Starting characters of field

Type options, press Enter.
  2=Edit          5=Display          14=Build          28=Check out
 29=Check in     30=Promote          39=Export ...

Opt Field      Record  Part      Group
19 QTY          PDFREC    PDF         MASTER

Command _____ Bottom
===>
F3=Exit          F4=Prompt      F5=Refresh       F9=Retrieve
F12=Cancel       F13=Repeat     F23=More options F24=More keys
This is a subsetting list.
```

Figure 43. Work with Fields Display in an Application Development Manager/400 session. This example shows the fields that match the criteria specified.



Tip:

In an Application Development Manager/400 session, AppDict Services/400 searches for the parts using the search path specified on the Start AppDict Services/400 (STRADS) display. Only the first occurrence of each field in the search path will be listed on the display.

3. To display information about the definition of a field, such as its length or description, type 5 (Display) next to its name.
4. To see which fields are directly or indirectly related to a specified field, from the Work with Fields display, type 19 (Work with field references) beside the field name, and press the Enter key.

The Work with Field References display appears. It lists all other fields in the dictionary that are related to the original field, and provides information about the reference level of each.

```

Work with Field References
Dictionary . . . . . : ADSLIB01DT
Field . . . . . : QTY          Record . . . . . : PDFREC
File . . . . . : PDF          Library . . . . . : AP01.MST

Type options, press Enter.
  2=Edit          5=Display          6=Design
 10=Work with programs referencing file  14=Compile

Opt  Level  Field      Record   File      Library  Attribute
—    -1    QTYB      REFREC   REFF      AP01.MST PF-DTA
—     0    QTY       PDFREC   PDF       AP01.MST PF-DTA
—     1    FLD002    ADF8R1   ADF8      AP01.MST DSPF
—     1    QTY       ADF8R1   ADF8      AP01.MST DSPF
—     1    QTY       PDFREC   PDFL      AP01.MST LF
—     2    EXPQTY    ADF4R1   ADF3      AP01.MST DSPF

Bottom
F3=Exit          F4=Prompt      F5=Refresh     F12=Cancel
F11=Display text F16=User options F23=More options F24=More keys

```

Figure 44. Work with Field References Display

```

Application Development Manager/400 Session

Work with Field References
Dictionary . . . . . : ADSLIB01DT      Part . . . . . : PDF
Field . . . . . : QTY          Record . . . . . : PDFREC
Group . . . . . : MASTER
Project . . . . . : ADMPRJ01

Type options, press Enter.
  2=Edit          5=Display          14=Build          28=Check out
 29=Check in     30=Promote ...

Opt  Level  Field      Part      Group
—    -1    QTYB      REFF      MASTER
—     0    QTY       PDF       MASTER
—     1    FLD002    ADF8      MASTER
—     1    QTY       ADF8      MASTER
—     1    QTY       PDFL      MASTER
—     2    EXPQTY    ADF3      MASTER

Bottom
F3=Exit          F4=Prompt      F5=Refresh     F10=Command entry
F12=Cancel      F16=User options F23=More options F24=More keys

```

Figure 45. Work with Field References Display in an Application Development Manager/400 Session

The *Level* prompt is interpreted as follows:

Level	Description
0	Indicates that this is the base field. In this example, the QTY field in the PDF file (or PDF part in an Application Development Manager/400 session) is the base field. The <i>Level</i> prompt, therefore, displays a value of 0 for this field and file.
1	Indicates that the field makes a direct reference to the base field. In this example, the ADF8 file (or ADF8 part in an Application Development Manager/400 session) has a FLD002 field that references the QTY field in PDF. The <i>Level</i> prompt, therefore, displays a value of 1 for the FLD002 field.
N (where N > 1)	Indicates that the field makes a reference to a field that is ranked at level N-1. In this example, the FLD002 field in the ADF8 file (ADF8 part in an Application Development Manager/400 session) is referenced by the EXPQTY field in the ADF3 file (ADF3 part in an Application Development Manager/400 session). The <i>Level</i> prompt, therefore, displays a value of 2 for the EXPQTY field.
-N (where N >= 1)	Indicates that the field is referenced by a field that is ranked at level -(N-1). In this example, the QTY field in the PDF file (PDF part in an Application Development Manager/400 session) references the QTYB field in the REFF file (REFF part in an Application Development Manager/400 session). The <i>Level</i> prompt, therefore displays a value of -1 for the QTYB field in the REFF file.



Tip:

The displays in Figure 44 on page 73 and Figure 45 on page 73 show the field references relative to a specific base field selected from the Work with Fields display. Fields with a level greater than 0 are derived from the base field; fields with a level of -1 are fields from which the base field is itself derived.

A field with a level of -1 may have other fields derived from it that are not shown on this display. Further, these fields may themselves be derived from other fields.

If you intend to edit a field with level -N, make it the base field on the Work with Fields display to ensure that you are seeing all of its derivations. For example, the QTYB field in the REFF file (REFF part, if you are working in an Application Development Manager/400 session) has a level -1 on the Work with Field References display in Figure 44 on page 73.

5. To make it the base field on the Work with Fields display:
 - a. Press F3 (Exit) on the Work with Field References display to return to the Work with Fields display.

- b. Type 19 (Work with field references) beside the QTYB field in REFF.
 Figure 46 illustrates the concept of a field reference level hierarchy:

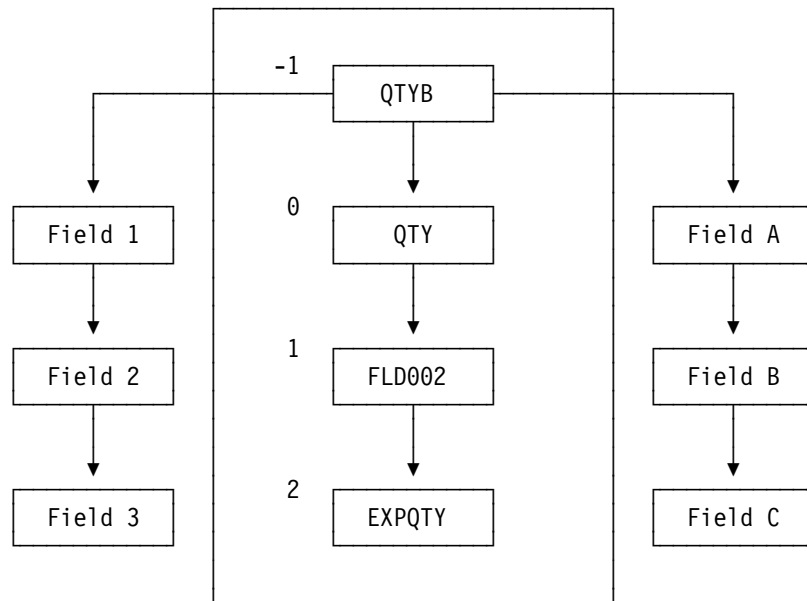


Figure 46. Relative Reference Levels of Fields

Each of the labeled boxes represents a field definition. The fields enclosed by the larger box represent the fields shown on the Work with Field References display, when the QTY field is selected as the base field. The connecting lines that run outside the (larger) box point to other fields (not shown on the Work with Field References display) that may be based on the QTYB field.

Determining Which Files Are Affected By a Change

To determine which files would be affected by a change to a field:

1. From the Application Dictionary Services/400 menu, type 1 (Work with fields), and press the Enter key.
 The Subset Fields to Work With display appears.
2. Fill in the fields as follows, and press the Enter key:
 - a. In the *Field* prompt, indicate the fields in the dictionary with which you want to work. In this example, type QTY.
 - b. In the *File* or *Part* prompt, type PDF.
 - c. In the other prompts, type *ALL.

The Work with Fields display appears, listing the fields you requested.

3. Type 8 (Work with impacted files) beside the QTY field, and press the Enter key.

The Work with Impacted Files display appears, listing all the files documented in the dictionary that will be affected by a change to this field.



Tip:

In an Application Development Manager/400 session, only the affected parts that are documented in the dictionary are listed. The specified search path option does not affect this list.

```

                                Work with Impacted Files
Dictionary . . . . . : ADSLIB01DT
Field . . . . . : QTY                Record . . . . . : PDFREC
File . . . . . : PDF                Library . . . . . : AP01.MST

Type options, press Enter.
  2=Edit                4=Delete                9=Work with related files
 10=Work with programs 14=Compile

Opt  Level  File      Library  Attribute  Text
---  ----  ---      ---      ---        ---
---   0    PDF       AP01.MST PF-DTA     Product File
---   1    ADF8     AP01.MST DSPF       ORDER PDNO ENTRY
---   1    PDFL     AP01.MST LF        PRODUCT FILE
---   1    TXFLA    AP01.MST LF-JOIN   ORDER TXF FILE (JOIN LF)
---   2    ADF3     AP01.MST DSPF       TXF Maintenance

                                Bottom
F3=Exit      F4=Prompt    F5=Refresh  F6=Work with objects to recreate
F12=Cancel   F21=Print list F24=More keys
```

Figure 47. Work with Impacted Files Display. This example shows the files that would be impacted.

Application Development Manager/400 Session

```

                                Work with Impacted Files
Dictionary . . . . . : ADSLIB01DT      Part . . . . . : PDF
Field . . . . . : QTY                  Record . . . . . : PDFREC
Group . . . . . : MASTER
Project . . . . . : ADMPRJ01

Type options, press Enter.
  2=Edit          4=Delete          9=Work with related files
 14=Build        28=Check out       29=Check in          30=Promote ...

Opt Level Part      Attribute Group
---   ---    ---
  ---   0    PDF        PF-DTA      MASTER
  ---   1    ADF8      DSPF        MASTER
  ---   1    PDFL      LF          MASTER
  ---   1    TXFLA     LF-JOIN     MASTER
  ---   2    ADF3      DSPF        MASTER

                                Bottom
F3=Exit          F4=Prompt    F5=Refresh      F6=Recreate parts
F10=Command entry F12=Cancel   F23=More options F24=More keys

```

Figure 48. Work with Impacted Files Display in an Application Development Manager/400 session. This example shows the files that would be impacted.

The *Level* prompt indicates the reference level for each file listed on the display. The reference level is used to determine the order in which the objects are compiled. It is interpreted as follows:

Level Description

- 0** Indicates that the selected field is defined in this file.
- 1** Indicates that a field in this file refers to the given field.
- N** Indicates that a field in this file refers to another field in the file that was previously defined as level N-1.

If a file has multiple reference levels in another file, the highest field reference level will be shown.

Example

As an example, consider the following field definition:

- Physical file F0 contains two fields, FLD0A and FLD0B, in record format RCD0.
- Physical file F1 contains the following fields in record format RCD1:

field FLD1A	REFFLD(RCD0/FLD0A F0)
field FLD1B	REFFLD(RCD0/FLD0B F0)
- Physical file F2 contains the following fields in record format RCD2:

field FLD2A	REFFLD(RCD1/FLD1A F1)
field FLD2B	REFFLD(RCD1/FLD1B F1)
- Physical file F3 contains the following fields in record format RCD3:

field FLD3A	REFFLD(RCD2/FLD2A F2)
field FLD3B	REFFLD(RCD1/FLD1A F1)

With respect to field FLD0A in file F0:

Field FLD3A has a field reference level of 3
Field FLD3B has a field reference level of 2

If the attribute of field FLD0A is changed, then in order to reflect changes in file F3, the following recompile sequence must be performed:

- 0 Recompile file F0
- 1 Recompile file F1
- 2 Recompile file F2
- 3 Recompile file F3

Therefore, the reference level of file F3 with respect to file F0 is the highest field reference level.

Determining Which Programs, Service Programs, and Modules Are Affected

After determining which files have been affected, you can determine which programs, service programs, or modules may need to be recompiled if any changes are made to a field.

1. From the Work with Impacted Files display, press F15 (Work with impacted programs).

The Work with Impacted Programs and Modules display appears, listing all the programs, service programs, and modules that refer to the files listed on the previous display.



Tip:

In an Application Development Manager/400 session, if the impacted program parts are not documented in the dictionary, they will not be listed. The specified search path option does not affect this list.

```

Work with Impacted Programs and Modules

Dictionary . . . . . : ADSLIB01DT
Field . . . . . : QTY
File . . . . . : PDF
Record . . . . . : PDFREC
Library . . . . . : AP01.MST

Type options, press Enter.
  2=Edit          4=Delete          14=Compile
 15=Scan RPG source      25=Find string

Opt Object      Library  Type      Attribute  Use  Text
— APDFI1      AP01.MST *PGM      RPG        01  Product data inquiry
— CBLC        AP01.MST *PGM      CBLLE      01  COBOL CALL C
— CHGXXXF     AP01.MST *SRVPGM   08
— RPT01       AP01.MST *PGM      RPG        01  RPG CALL CLP TEST
— TT8         AP01.MST *MODULE   CLLE       03  DATAQ CALLING PGM
— YEH1        AP01.MST *MODULE   CLLE       08  Call yehcalling
— YEH2        AP01.MST *PGM      CLP        08  call yehcalling

Bottom
F3=Exit          F4=Prompt          F5=Refresh
F6=Work with objects to recreate  F12=Cancel        F24=More keys

```

Figure 49. Work with Impacted Programs and Modules Display. This example shows the programs, service programs, and modules that would be impacted.

```

Application Development Manager/400 Session

Work with Impacted Programs and Modules

Dictionary . . . . . : ADSLIB01DT
Field . . . . . : QTY
Group . . . . . : MASTER
Project . . . . . : ADMPRJ01
Part . . . . . : PDF
Record . . . . . : PDFREC

Type options, press Enter.
  2=Edit          4=Delete          14=Build          15=Scan RPG source
 25=Find string  28=Check out     29=Check in      30=Promote

Opt Part      Type      Attribute  Use  Group
— APDFI1      MODULE   RPGLE      01  MASTER
— CBLC        MODULE   CBLLE      01  MASTER
— CHGXXXF     SRVPGM   08  MASTER
— INVRPGA     PGM      RPG        03  MASTER
— RPT01       PGM      RPG        01  MASTER
— TT8         SRVPGM   03  MASTER
— YEH1        MODULE   CLE        08  MASTER

More...
F3=Exit          F4=Prompt          F5=Refresh
F10=Command entry  F12=Cancel        F6=Recreate parts
F24=More keys

```

Figure 50. Work with Impacted Programs and Modules Display in an Application Development Manager/400 Session. This example shows the programs, service programs, and modules that would be impacted.

For information about the *Use* prompt, see “Finding Programs, Service Programs, or Modules That Reference a File” on page 102.

Changing a Field

The following example shows you how to change a field definition. The field in this example is called QTY.

1. From the Application Dictionary Services/400 menu, type 1 (Work with fields), and press the Enter key.

The Subset Fields to Work With display appears.

2. Fill in the fields as follows, and press the Enter key:
 - a. In the *Field* prompt, type the name of the field that you want to change (QTY in this case).
 - b. In the *File* or *Part* prompt, type PDF.
 - c. In the other prompts, type *ALL.

The Work with Fields display appears.

3. Type 2 (Edit) beside the field you want to change, and press the Enter key.

The Source Entry Utility (SEU) Edit display appears. Here, you can make changes to the source member of the object.

Note: The source file of an object will not be documented in the dictionary, because the AppDict Services/400 feature will determine this.

4. Apply some change to the field QTY; for example, change its length. (Refer to the *ADTS/400: Source Entry Utility* guide for information about how to edit source members.)
5. Press F3 (Exit) to go to the SEU Exit display.
6. Press the Enter key to save the member.

The Work with Fields display appears.

7. Now that the field is changed, you need to recompile the file that contains it. There are two ways of doing this:

- Recompile only the file that contains the changed field.

To do this, type 14 (Compile), or in an Application Development Manager/400 session, type 14 (Build), next to the field you have just changed.

Note: You do not need to do this if you have specified Y in the *Compile after editing* prompt on the Change Defaults display (as shown in Figure 7 on page 20).

If Y is specified, the Create File display appears (or, for an Application Development Manager/400 session, the BLDPART display appears) when you exit the SEU editor.

- Recreate all impacted objects, including not only the file that contains the changed field, but also all other objects that are impacted by the change to the field.

Recreating Impacted Objects

This section describes how to recreate objects that have been affected by a change to a field or file.

AppDict Services/400 will not list the programs, service programs, and modules that are impacted by changes to other programs, service programs, or modules, but it will list programs based on other criteria, such as the calling hierarchy and ILE binding relationships. Therefore, you can also use AppDict Services/400 to recreate or recompile affected programs, service programs, or modules that you have listed using a method *other than 8* (Work with impacted files) or F15 (Work with impacted programs).

After you have changed an object (field, file, or program), you can recreate all of the other objects affected by the change.

This section describes:

- How the recreate function works
- How to recreate from directly within AppDict Services/400
- How to recreate using AppDict Services/400 from within an Application Development Manager/400 session

How the Recreate Function Works

There are two aspects to understanding how the recreate function works:

- How AppDict Services/400 decides which default options to use
- The order in which parts are recreated
- The results of a failure in the recreation process

Which Default Options Does AppDict Services/400 Use

The F6 (Submit job to recreate) uses default options as follows:

- If you are recreating a program, service program, or module that you previously created in AppDict Services/400, the parameters that you used earlier are used again.
- If you are recreating a program, service program, or module that is documented in AppDict Services/400 but was not previously created within AppDict Services/400, AppDict Services/400 retrieves the parameters from the documented program, service program, or module.

Exceptions

From the documented program, service program, or module AppDict Services/400 will not retrieve any special options that are not the defaults, such as:

1. Entry module (ENTMOD(*FRIST) is the default)
2. Binding directory (*BNDDIR)
3. Export source file (EXPORT (*ALL) is the default)

However, if you prompt on the rebuild action and change the options, AppDict Services/400 stores the new build command and reuses it for future builds.

Order in Which AppDict Services/400 Recreates Parts

When AppDict Services/400 recreates OPM program parts, it recreates the program building blocks, such as files and called programs first, then it recreates the larger program.

When AppDict Services/400 recreates program parts, it recreates them in a specific order:

1. After any logical files have been deleted, the physical files are recreated.
2. The logical files are recreated.
3. The service programs are recreated.
4. Finally, the ILE programs are recreated, and the OPM programs are recompiled.

Results of a Failure in the Recreation Process

Because of the order in which AppDict Services/400 recreates program objects, if a failure occurs during the recreation of physical files, data can be lost. When AppDict Services/400 recreates all of the objects in a program, it handles logical and physical files as follows:

1. First it examines all of the objects in the list of objects to recreate.
2. Next it deletes the logical files.

Because physical files cannot be deleted or replaced if there are logical files based on them, the logical files have to be deleted before the physical files can be recreated.

3. Then it recreates the physical files.
4. Next it recreates the logical files.
5. Finally, it recreates the rest of the program parts.

If a failure occurs during the recreation of any of the physical files, AppDict Services/400 stops. For example, a failure will occur if a field type in a physical file is changed from character to numeric. Failures are logged in the spooled file.

In this example, if the failure occurs before the logical files can be recreated, you will have to recreate them yourself. Depending on the time that passes between the time the logical files were deleted and the time the failure occurs, the logical files may also be deleted from the dictionary. If this happens, you will have to recreate the logical files from outside AppDict Services/400.

Recreating Directly in AppDict Services/400

This section describes the two most common situations when you will want to use the recreate function:

- When changes have been made to OPM or ILE fields or files
- When changes have been made to ILE modules or service programs

When Changes Have Been Made to Fields or Files

After you have made changes to fields or files, the following example shows you how to recreate impacted objects:

1. If you have made a change to *field*, go to the Work with Objects to Recreate display in one of the following ways. (If you are recreating objects because you have made a change to a file, do not complete this step; go to step 2.)

Note: It does not matter which of the following methods you use. The results will be the same. Each one provides you with a different view of the impacted objects along the way.

- From the Work with Fields display, type 20 (Work with objects to recreate) beside the field or file you have changed, and press the Enter key. (This example uses the field QTY of file PDF.)

The Work with Objects to Recreate display appears.

- From the Work with Fields display:
 - a. Beside the field that you changed, type 8 (Work with impacted files), and press the Enter key.

The Work with Impacted Files display appears.

- b. Press F6 (Work with objects to recreate).

The Work with Objects to Recreate display appears.

- From the Work with Fields display:
 - a. Beside the field that you changed, type 8 (Work with impacted files), and press the Enter key.

The Work with Impacted Files display appears.

- b. Press F15 (Work with impacted programs).

The Work with Impacted Programs and Modules display appears.

- c. Press F6 (Work with objects to recreate).

The Work with Objects to Recreate display appears.

Note: You can also get to the Work with Objects to Recreate display by pressing F6 (Work with objects to recreate) from the Work with Fields in Record Format display.

2. If you made a change to a *file*, go to the Work with Objects to Recreate display as follows:

- a. From the Work with Files display, type option 20 (Work with objects to recreate), and press the Enter key.

The Work with Objects to Recreate display appears.

```

Work with Objects to Recreate

Dictionary . . . . . : ADSLIB01DT
Object . . . . . : PDF                               Type . . . : *FILE
Library . . . . . : AP01.MST

Type options, press Enter.
  2=Edit                               4=Drop
 16=Change compile command

Opt Object      Library  Type      Attribute  Text
---
ADF3      AP01.MST  *FILE     DSPF       TXF Maintenance
ADF8      AP01.MST  *FILE     DSPF       ORDER PDNO ENTRY
APDFI1    AP01.MST  *PGM      RPG        Product data inquiry
CBLC      AP01.MST  *PGM      CBL        COBOL CALL C
PDF       AP01.MST  *FILE     PF-DTA     Product File
PDFL      AP01.MST  *FILE     LF         PRODUCT FILE
RPT01     AP01.MST  *PGM      RPG        RPG CALL CLP TEST
TT8       AP01.MST  *PGM      CLP        DATAQ CALLING PGM
More...

F3=Exit      F4=Prompt      F5=Refresh      F6=Submit recreate job
F11=Display more text  F12=Cancel      F24=More keys

```

Figure 51. Work with Objects to Recreate Display. This example shows the objects that are to be created.

- To submit a batch job to recreate the objects in their libraries, press F6 (Submit recreate job).

The information about the objects in the dictionary (ADSLIB01DT, in this case) is updated.

All the affected objects shown on this display are recreated.

To change the options used by the create command for the object, you can:

- Type 16 (Change compile command).

The Change Compile Command display appears.

- Fill in the prompts, and press the Enter key.

The AppDict Services/400 feature uses these options the next time you recreate the object.



Tips:

- If you do not have authority to the user profile or the job description of the objects listed on the Work with Objects to Recreate display, the Confirm Recreation of Objects display will appear to warn you. You must either get authority from your system administrator, or type option 4 (Drop) to drop the objects that you do not have authority to recreate.

If the object creation is not successful, the information about the object in the dictionary will not be updated.

Use F19 (Change submit job) to change the attribute of the job submitted when you use F6 (Submit recreate job). See the online help for information about the reports generated after the recreation job has been completed.

- The object which is impacted but not documented in AppDict Services/400 will not appear on this display. This may result in the failure of the job submitted.

When Changes Have Been Made to ILE Modules or Service Programs

After you have made a change to an ILE module or service program, you can recreate the ILE programs and service programs that bind that module, or recreate all of the ILE programs that bind that service program as follows:

1. From the AppDict Services/400 menu, select option 3 (Work with programs and modules), and press the Enter key.

The Work with Programs and Modules display appears.

2. Recompile the module or recreate the service program that you changed by typing option 14 (Compile) beside the module or service program that you changed, and pressing the Enter key.

3. Beside the module or service program that you changed, type option 16 (Work with binding programs), and press the Enter key.

The Work with Binding Programs display appears showing all of the ILE programs and service programs that bind the selected module, or all of the ILE programs that bind the selected service program.

4. Beside the first object listed, type option 14 (Compile).

5. Press F13 (Repeat) to put option 14 (Compile) beside the rest of the objects listed in the Work with Binding Programs display, and press the Enter key.

Jobs are submitted to recreate all of the ILE programs and service programs.

Recreating from Within an Application Development Manager/400 Session

In an Application Development Manager/400 session, type 14 (Build), beside the changed object (in the Work with Fields, Work with Files, or Work with Programs and Modules display), and press F4 to specify *EXTEND for the SCOPE parameter to rebuild all of the parts that have been affected by the change.

Note: The authority for the newly created file parts are in the user profile called QPRJOWN. You may have authority problems if you work with the object parts when you are not in an Application Development Manager/400 session.

Chapter 8. Finding Objects and Their Relationships to Other Objects

This chapter describes how to find various types of objects and their relationships to other objects. It describes how to find:

- Strings, using the find string and scan RPG source functions
- Programs, service programs, and modules through their calling structure and references to data areas and files
- Files, through their relationship to other files and to programs, service programs, and modules
- Record formats defined in files
- Data areas, by searching for their use in programs, service programs, and modules
- OPM program and ILE module objects, by their last changed source date

Finding Strings and Scanning ILE Program Objects

Because of the other functions that AppDict Services/400 offers, you should not have to use the find and scan options as often as you would have without AppDict Services/400. For example, to find out which ILE programs or service programs bind a particular module, you can use option 16 (Work with binding programs) instead of searching for the module name in your source code. Also, the find string and scan options can only be used with a subset of program objects:

- Option 25 (Find string) can only be used with OPM source members or ILE modules
- Option 15 (Scan RPG source) can only be used with RPG source members with a record length of 92 bytes

How to Find Strings and Scan RPG Source

The Application Dictionary Services/400 (AppDict Services/400) feature uses AS/400 system cross-reference information which only provides information about where a record format is used in a program, and not whether a specific field in a record is used in a program. AppDict Services/400, therefore, cannot list the fields used in a program, but it allows you to search for a specific field name using the 25 (Find string) option. This option is provided on the following displays:

- Work with Programs and Modules
- Work with Impacted Programs and Modules
- Work with Programs Referencing File
- Work with Calling Programs
- Work with Called Programs
- Work with Programs Referencing a Data Area

For RPG programs, the AppDict Services/400 feature also provides the Scan RPG source function, which permits you to search for externally described fields. If the scanned field is renamed in the input specification of the program, you can use the 15 (Scan RPG source) function to search for occurrences of both the original field name and the new field name in the RPG source member, starting from the column

number that you specified on the Scan Source of RPG Program display. The 15 (Scan RPG source) function is also provided on the Work with Impacted Programs and Modules and the Work with Programs Referencing File displays.

Note: Option 15 (Scan RPG source) can only be used with RPG source members with a record length of 92 bytes.

This section shows you, through examples, how to:

- Search for strings in source members
- Search for strings in RPG programs, including externally described field names and program-described field names in RPG data structures.

Finding Strings in Source Members

To search for a string in the source members of OPM programs and modules:

1. From the Application Dictionary Services/400 menu, type 3 (Work with programs and modules) to get to the Subset Programs and Modules to Work With display.
2. Specify the subsetting options needed, and press the Enter key to proceed to the Work with Programs and Modules display.
3. Type 25 (Find string) beside the OPM programs and modules you want to search, and press the Enter key.

If you want to search for the string in all the programs listed, press F13 (Repeat).

The Find String Using PDM display appears. All the option prompts will be filled in with the number 25.

Find String Using PDM		
Type choices, press Enter.		
Find	CUSREC	
From column number	<u>1</u>	1-*RCDLEN
To column number	<u>*RCDLEN</u>	1-*RCDLEN
Kind of match	<u>2</u>	1=Same case 2=Ignore case
Option	<u>5</u>	*NONE, Valid option
Prompt	<u>N</u>	Y=Yes, N=No
Print list	<u>N</u>	Y=Yes, N=No
Print records	<u>N</u>	Y=Yes, N=No
Number to find	<u>*ALL</u>	*ALL, number
Print format	<u>*CHAR</u>	*CHAR, *HEX, *ALTHEX
Mark record	<u>Y</u>	Y=Yes, N=No
Record overflow	<u>1</u>	1=Fold, 2=Truncate
Find string in batch	<u>N</u>	Y=Yes, N=No
Parameters		

F3=Exit	F5=Refresh	F12=Cancel

Figure 52. Find String Using PDM Display

4. Fill in the prompts as follows, and press the Enter key:
 - a. In the *Find* prompt, type the string you want to find (CUSREC in this case).
 - b. In the *Option* prompt, type 5 to display the source member that contains the search string.

The SEU Edit display appears, with the cursor on the first character of the string that you want to find.

Note: In an Application Development Manager/400 session, the FNDSTRPART command is used instead of the FNDSTRPDM command.

Finding a Field Name in an RPG Program

To find field names (including externally-described fields) in an RPG program:

1. From the Application Dictionary Services/400 menu, type 1 (Work with fields), and press the Enter key.

The Subset Fields to Work With display appears.

2. Fill in the prompts as follows, and press the Enter key:

- a. In the *Field* prompt, type the name of a field in the RPG input specification.
- b. In the other prompts, type *ALL.

The Work with Fields display appears.

3. Type 10 (Work with programs referencing file) beside the field you want to change, and press the Enter key.

The Work with Programs Referencing File display appears.

4. In the *Opt* prompt, type 15 (Scan RPG source) beside the program you want to scan, and press the Enter key.

The Scan Source of RPG Program display appears.

```
Scan Source of RPG Program
Dictionary . . . . . : ADSLIB01DT
Type choices, press Enter.
Scan text . . . . . : FLDNAM
Start scan in column . . . . . : 1 1-80
F3=Exit      F5=Refresh      F12=Cancel
```

Figure 53. Scan Source of RPG Program Display

5. Fill in the prompts as follows, and press the Enter key:

- a. In the *Scan text* prompt, type the string that you want to scan (FLDNAM in this case). This string can only be up to 10 characters.
- b. In the *Start scan in column* prompt, specify the starting position (1 in this case), and press the Enter key to submit the RPG source scan batch job.

The AppDict Services/400 feature scans for all occurrences of the specified text string, starting from the column specified.

A message will appear when the source scanning job has completed. You can use the WRKSPLF command to display the spooled file that contains the job's result.

Finding Programs, Service Programs, and Modules

You can find programs, service programs, and modules by:

- Looking at their calling structure
- Finding data areas that they refer to
- Finding files that they refer to

Determining the Program Calling Structure

This section describes:

- How the calling structure is defined in the Integrated Language Environment
- How to determine the calling structure for:
 - OPM programs and procedures
 - ILE programs and objects

How is the Calling Structure Defined in the Integrated Language Environment

Under OPM, programs were written and compiled. The only mechanism of linking programs to other programs was the program call. The program call that OPM programs use exclusively is the dynamic program call. OPM programs can call programs written in the same language as the calling program or a different language. They cannot, however, combine more than one language in a single program.

In ILE, you can call either a program or a procedure, but you must indicate whether the program object you are calling is a program or a procedure. To do this, you use a separate call statement to represent each type of call. Programs are called dynamically and procedures are called statically.

A **dynamic program call** transfers control to either an ILE program object or an OPM program object. Restrictions apply to which ILE program objects can make valid dynamic program calls. OPM programs, ILE programs, and service programs can make dynamic program calls to other OPM programs or ILE programs only. Service programs cannot be dynamically called.

A **static procedure call** transfers control to an ILE procedure. Static procedure calls can be coded in ILE languages only. A static procedure call can be used to call any of the following:

- A procedure within the same module
- A procedure within a separate module within the same ILE program or service program
- A procedure that has been exported from an ILE service program

Note: ILE procedure calls are not supported by AppDict Services/400.

The AppDict Services/400 feature allows you to determine the calling structure for OPM programs and for ILE programs, service programs, and modules. Determining the calling structure can be useful in assessing the affect of proposed changes, or when you are working with the logic of larger programs. This section describes how to determine the calling structure for both of these types of programs.

Determining the Calling Structure for OPM Programs and Procedures

OPM programs and procedures can be called by other OPM programs, System/36 OCL procedures, or service programs; likewise, OPM programs, procedures, and System/36 OCL procedures can call other OPM programs or procedures.

In AppDict Services/400, you use the Work with Calling Programs display to list the programs, service programs, and System/36 OCL procedures that call the program or procedure that you specify on the Work with Programs and Modules display. You use the Work with Called Programs display to list the programs or procedures that are called by the program, service program, or System/36 OCL procedure that you specified on the previous display. This section describes how to use these displays. The example used in this section illustrates how to find out the calling structure for an OPM program that refers to a changed field.

To determine the calling structure for OPM programs and procedures:

1. From the Application Dictionary Services/400 menu, type 3 (Work with programs and modules), press the Enter key.

The Subset Programs and Modules to Work With display appears.

Subset Programs and Modules to Work With

Dictionary : ADSLIB01DT

Type choices, press Enter.

Object	<u>YEH1</u>	*ALL, name, generic*
Library	<u>*ALL</u>	*ALL, name, generic*
Type	<u>*ALL</u>	*ALL,*PGM,*SRVPGM,*MODULE
Attribute	<u>*ALL</u>	*ALL, attribute
		F4 for list

Text:

Search words _____

Search condition 1 1=Or, 2=And

F3=Exit F4=Prompt F5=Refresh F12=Cancel

Figure 54. Subset Programs and Modules to Work With Display

Application Development Manager/400 Session

```

                Subset Programs and Modules to Work With

Dictionary . . . . . : ADSLIB01DT
Project . . . . . : ADMPRJ01

Type choices, press Enter.

Part . . . . . YEH1      *ALL, name, generic*
Type . . . . . *ALL      *ALL, PGM, SRVPGM, MODULE
Attribute . . . . . *ALL  *ALL, attribute
                                F4 for list

Text:
Search words . . . . . _____
                                _____

Search condition . . . . . 1  1=Or, 2=And

F3=Exit      F4=Prompt      F5=Refresh      F12=Cancel
    
```

Figure 55. Subset Programs and Modules to Work With Display in an Application Development Manager/400 Session

2. Fill in the prompts as follows, and press the Enter key:
 - a. In the *Object* or *Part* prompt, type the name of the program you want to work with (YEH1 in this case).
 - b. In the other prompts, type *ALL.

The Work with Programs and Modules display appears.

```

                Work with Programs and Modules

Dictionary . . . . . : ADSLIB01DT
Position to . . . . . _____ Starting characters of program

Type options, press Enter.
 2=Edit          3=Copy          4=Delete          5=Display
 6=Work with calling programs  8=Work with called programs ...

Opt Object      Library  Type  Attribute  Text
8  YEH1        AP01.MST *PGM  CLLE      Call yehcalling

                                                Bottom

Command
====>
F3=Exit      F4=Prompt      F5=Refresh      F6=Create
F9=Retrieve  F12=Cancel     F23=More options F24=More keys
This is a subsetting list.
    
```

Figure 56. Work with Programs and Modules Display. This example shows the program with which you want to work.

Application Development Manager/400 Session

```
Work with Programs and Modules

Dictionary . . . . . : ADLIB01DT
Position to . . . . . : _____ Starting characters of part

Type options, press Enter.
  2=Edit          3=Copy          4=Delete          5=Display
  9=Run           14=Build         28=Check out     29=Check in...

Opt Part Type Attribute Group
8_ YEH1 PGM CLP MASTER

Command _____ Bottom
===>
F3=Exit          F4=Prompt          F5=Refresh          F6=Create
F9=Retrieve       F12=Cancel         F23=More options   F24=More keys
This is a subsetting list.
```

Figure 57. Work with Programs and Modules Display in an Application Development Manager/400 Session. This example shows the program with which you want to work.



Tips:

1. The Work with Programs and Modules display lists those programs and non-ILE procedures in the specified dictionary that meet the subsetting criteria. If you have a non-ILE procedure with the same name as a program in the same library, then only the last one that was documented will be displayed on the Work with Programs and Modules display. Therefore, if you want to display both the non-ILE procedure and the program with the same name, we recommend that you put the non-ILE procedure with the same name in a different library from the program.
2. In an Application Development Manager/400 session, the AppDict Services/400 feature searches for the parts using the search path that you specified on the Start AppDict Services/400 (STRADS) display. Only the first occurrence of each program in the search path will be listed on the display.
3. To find out if the changed field impacts *called* programs or procedures, complete step 3a. To find out if the changed field impacts *calling* programs, service programs, or System/36 OCL procedures, complete step 3b on page 95.
 - a. To obtain a list of programs and procedures that are called by a program, service program, or System/36 OCL procedure (in this case, YEH1), that you specified on the Work with Programs and Modules display:
 - 1) Type 8 (Work with called programs) beside the name of the calling program, service program, or System/36 OCL procedure (YEH1 in this case), and press the Enter key.

The Work with Called Programs display appears, listing all programs and procedures called by YEH1. In this case, only one program (YEHCALLED) is displayed.

```

Work with Called Programs

Dictionary . . . . . : ADSLIB01DT      Library . . . . . : AP01.MST
Program . . . . . : YEH1              Type . . . . . : *PGM
Called level . . . . . : +1           Attribute . . . . . : CLLE
Text . . . . . : Call yehcalled

Type options, press Enter.
  2=Edit      4=Delete      5=Display      14=Compile ...
  8=Work with called programs      9=Run

Opt Program Library Attribute Text
  _ YEHCALLED AP01.MST CLP Called program for YEH1 program

Bottom
F3=Exit      F4=Prompt      F5=Refresh      F10=Command entry
F12=Cancel   F16=User options F23=More options F24=More keys

```

Figure 58. Work with Called Programs Display. This example shows the program called by the selected program.

```

Application Development Manager/400 Session

Work with Called Programs

Dictionary . . . . . : ADSLIB01DT      Part . . . . . : YEH1
Called level . . . . . : +1           Type . . . . . : PGM
Group . . . . . : MASTER              Attribute . . . . . : CLLE
Project . . . . . : ADMPRJ01

Type options, press Enter.
  2=Edit      4=Delete      5=Display      9=Run
 14=Build    28=Check out 29=Check in   30=Promote ...

Opt Part Attribute Group
  _ YEHCALLED CLP MASTER

Bottom
F3=Exit      F4=Prompt      F5=Refresh      F10=Command entry
F12=Cancel   F13=Repeat    F23=More options F24=More keys

```

Figure 59. Work with Called Programs Display in an Application Development Manager/400 Session. This example shows the program called by the selected program.

The *Called level* prompt in the display indicates the level, within the calling hierarchy, of the called programs and procedures listed relative to the program, service program, or System/36 OCL procedure that you selected. This level is referred to as the **called level**.

For example, if the programs and procedures are called directly by a program that you selected, then the called level is +1. If you select option 8 (Work with called programs) beside a program with called level +1, then the called programs and procedures will have a called level of +2, and so on. The maximum called level is +200.

- 2) To determine if the field in question has been passed to the called program, type 2 (Edit) next to YEHCALLED to see its source.
- 3) Press F12 (Cancel) to return to the Work with Called Programs display.
- 4) Press F12 (Cancel) again to return to the Work with Programs and Modules display.

b. To find out if the changed field impacts programs, service programs, and System/36 OCL procedures that *call* the program you specified on this display (in this case, YEH1):

- 1) Type 6 (Work with calling programs) beside its name, and press the Enter key.

The Work with Calling Programs display appears, listing all of the programs, service programs, and System/36 OCL procedures that call YEH1. In this case, the program YEHCALLING is displayed.

```

                                Work with Calling Programs
Dictionary . . . . . : ADSLIB01DT
Program . . . . . : YEH1           Library . . . . . : AP01.MST
Calling level . . . . . : -1       Attribute . . . . . : CLLE
Text . . . . . : Call yehcalled

Type options, press Enter.
 2=Edit           4=Delete           5=Display
 8=Work with calling programs      9=Run           14=Compile ...

Opt Program Library Type Attribute Text
_ YEHCALLING AP01.MST *PGM RPGLE Calling program for YEH1

                                Bottom
F3=Exit           F4=Prompt           F5=Refresh           F10=Command entry
F12=Cancel        F16=User options    F23=More options    F24=More keys
  
```

Figure 60. Work with Calling Programs Display. This example shows the program that calls the selected program.

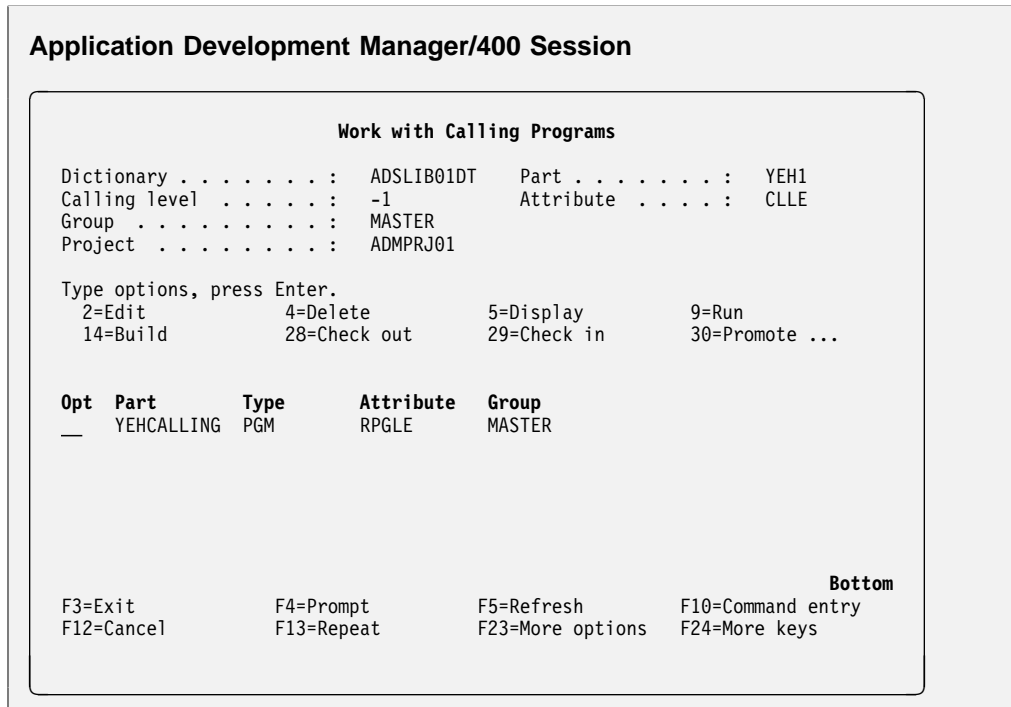


Figure 61. Work with Calling Programs Display in an Application Development Manager/400 Session. This example shows the program that calls the selected program.

The *Calling level* prompt in the display indicates the level, within the calling hierarchy, of the calling programs, service programs, and System/36 OCL procedures listed relative to the program or procedure that you selected. This level is referred to as the **calling level**.

For example, if the programs, service programs, or System/36 OCL procedures are directly calling the program that you selected, then the calling level is -1. If you select option 6 (Work with calling programs) beside a program with calling level -1, then the calling programs, service programs, or System/36 OCL procedures will have a calling level of -2, and so on. The maximum calling level is -200.

- 2) To determine if the field in question is being passed by the calling programs type 2 (Edit) next to the program to see its source.
- 3) Press F12 (Cancel) to return to the Work with Calling Programs display.
- 4) Press F12 (Cancel) again to return to the Work with Programs and Modules display.

Note: Not all AS/400 compilers supply information about the calling hierarchy of programs. This option only supports for BASIC, CL, COBOL, PL/I and RPG programs and OCL36 procedures, and the ILE equivalents of these program and procedure types. (One exception applies: program references are not available for CLE modules.)

Determining the Referencing Structure for ILE Programs and Objects

To determine if ILE objects reference other ILE objects, and to determine what those objects are:

1. From the Application Dictionary Services/400 menu, type 3 (Work with programs and modules), and press the Enter key.

The Subset Programs and Modules display appears.

2. Fill in the prompts as follows, and press the Enter key:

- a. In the *Object* or *Part* prompt, type the name of the program or part with which you want to work (YEHMOD, in this case).
- b. In the other prompts, type *ALL.

The Work with Programs and Modules display appears.

3. To display a list of ILE programs or service programs that reference a specific service program or module, complete step 3a. To display a list of modules or service programs that are bound into a specific ILE program or service program, complete step 3b on page 98.

- a. To obtain a list of ILE programs or service programs that reference a specific service program or module:

- 1) Type 16 (Work with binding programs) beside the name of the service program or module with which you want to work (YEHMOD1, in this case), and press the Enter key.

The Work with Binding Programs display appears.

Work with Binding Programs

Dictionary : ADSLIB01DT Library : AP01.MST
 Bound object : YEHMOD1 Type : *MODULE
 Reference level : -1 Attribute : CLLE
 Text : Bind yeh module

Type options, press Enter.
 4=Delete 5=Display 9=Run 14=Compile
 16=Work with binding programs...

Opt	Program	Library	Type	Attribute	Text
_	YEBINDING	AP01.MST	*PGM	RPGLE	Binding OBJ1

Bottom

F3=Exit F4=Prompt F5=Refresh F10=Command entry
 F12=Cancel F16=User options F23=More options F24=More keys

Figure 62. Work with Binding Programs Display. This example shows the ILE program that references the specified module.

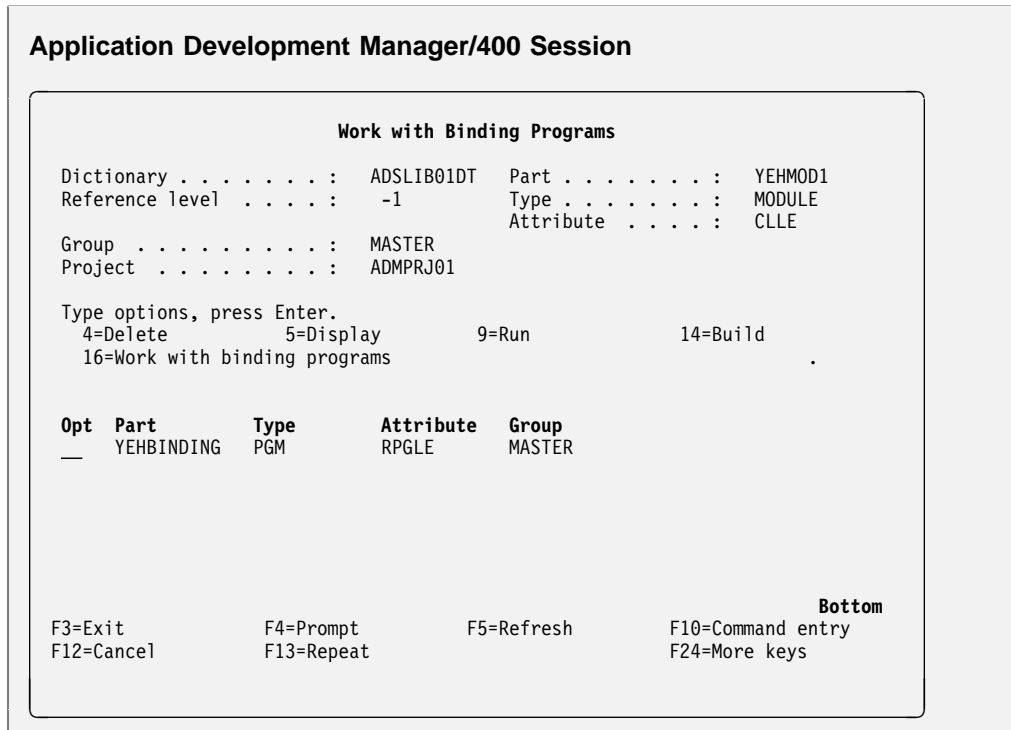


Figure 63. Work with Binding Programs Display in an Application Development Manager/400 Session. This example shows the ILE program that references the specified module.

- b. To obtain a list of modules and service programs that are bound into a specified ILE program or service program:
 - 1) Type 19 (Work with bound objects) beside the ILE program or service program with which you want to work (YEHOBJ1, in this case), and press the Enter key.
The Work with Bound Objects display appears.
 - 2) To determine if the particular parameter has been passed between ILE programs or service programs, type 2 (Edit) beside the referencing programs, and press the Enter key.

Note: You cannot edit service programs because they may have multiple source files, and you cannot edit ILE programs because they are not associated with any source files.

```

Work with Bound Objects
Dictionary . . . . . : ADSLIB01DT      Library . . . . . : AP01.MST
Program . . . . . : YEH0BJ1          Type . . . . . : *PGM
Reference level . . . . . : +1          Attribute . . . . . : CLLE
Text . . . . . : Reference yeh object

Type options, press Enter.
  2=Edit          4=Delete          5=Display          14=Compile
  19=Work with bound objects...

Opt Object      Library  Type      Attribute  Text
—  YEHBOUND     AP01.MST *SRVPGM
—  YEHMOD       AP01.MST *MODULE      Bound program for YEH0BJ1
                               Entry module for YEH0BJ1

Bottom
F3=Exit          F4=Prompt      F5=Refresh      F10=Command entry
F12=Cancel       F16=User options  23=More options  F24=More keys

```

Figure 64. Work with Bound Objects Display. This example shows the service program that is bound into the selected ILE program.

```

Application Development Manager/400 Session

Work with Bound Objects
Dictionary . . . . . : ADSLIB01DT      Part . . . . . : YEH0BJ1
Reference level . . . . . : +1          Type . . . . . : PGM
                                          Attribute . . . . . : CLLE

Group . . . . . : MASTER
Project . . . . . : ADMPRJ01

Type options, press Enter.
  2=Edit          4=Delete          5=Display          30=Promote ...
  14=Build        28=Check out     29=Check in

Opt Part      Type      Attribute  Group
—  YEHBOUND     SRVPGM
—  YEHMOD       MODULE      MASTER
                               MASTER

Bottom
F3=Exit          F4=Prompt      F5=Refresh      F10=Command entry
F12=Cancel       F13=Repeat     F23=More options  F24=More keys

```

Figure 65. Work with Bound Objects Display in an Application Development Manager/400 Session. This example shows the service program that is bound into the selected ILE program.

3) To determine if the particular parameter has been passed between ILE modules or service programs, type 2 (Edit) beside the referenced objects, and press the Enter key.

Note: You cannot edit service programs because they may have multiple source files.

Finding Programs Referencing a Data Area

To find programs, service programs, or modules that reference a certain data area:

1. From the Application Dictionary Services/400 main menu, type 4 (Work with other objects), and press the Enter key.

The Subset Objects to Work With display appears.

Subset Objects to Work With

Dictionary : ADSLIB01DT

Type choices, press Enter.

Object	<u>*ALL</u>	*ALL, name, generic*
Library	<u>*ALL</u>	*ALL, name, generic*
Type	<u>*DTAARA</u>	*ALL, *type F4 for list

Text:

Search words _____

Search condition 1 1=Or, 2=And

F3=Exit F4=Prompt F5=Refresh F12=Cancel

Figure 66. Subset Objects to Work With Display

2. Fill in the prompts as follows, and press the Enter key:
 - a. In the *Object* and the *Library* prompts, type *ALL.
 - b. In the *Type* prompt, type *DTAARA.

The Work with Other Objects display appears.

```

Work with Other Objects
Dictionary . . . . . : ADSLIB01DT
Position to . . . . . _____ Starting characters of file
Type . . . . . _____

Type options, press Enter.
  2=Edit          3=Copy          4=Delete          5=Display
  7=Rename        10=Work with programs referencing data area ...

Opt Object      Library   Type      Text
—   DTAARA        ADSLIB01DT *DTAARA   F6 Create
—   DTAARA1       ADSLIB01DT *DTAARA   Dictionary space Analysis & Reorg
—   DTAARA2       ADSLIB01DT *DTAARA   Control information on dictionary reorg
—   QDNDARGZ      ADSLIB01DT *DTAARA   Control information on dictionary reorg
—   QDNDARTL      ADSLIB01DT *DTAARA   DICTIONARY space Analysis & Control

Bottom
Command
====>
F3=Exit          F4=Prompt      F5=Refresh      F6=Create
F9=Retrieve      F12=Cancel     F23=More options F24=More keys
This is a subsetted list.

```

Figure 67. Work with Other Objects Display

- Type 10 (Work with programs referencing data area) (or, if you are working in an Application Development Manager/400 session, type 10 (Work with parts referencing data area)) beside the name of the object (DTAARA in this case) for which you want to see the referencing programs, and press the Enter key.

The Work with Programs Referencing Data Area display appears (or, if you are working in an Application Development Manager/400 session, the Work with Parts Referencing Data Area display appears), listing all of the programs, service programs, or modules referencing the DTAARA data area. (In this case there are two programs referenced, ATXFD11 and CPGM4.)

```

Work with Programs Referencing Data Area
Dictionary . . . . . : ADSLIB01DT
Data area . . . . . : DTAARA          Library . . . . . : ADSLIB01DT
Text . . . . . : F6 Create

Type options, press Enter.
  2=Edit          4=Delete          14=Compile
  18=Display object information  25=Find string

Opt Program    Library   Type      Attribute  Text
—   ATXFD11     ADSLIB01DT *PGM      RPG        TXF Maintenance (Data Area test)
—   CPGM4       ADSLIB01DT *PGM      CLP        test

Bottom
F3=Exit          F4=Prompt      F5=Refresh      F10=Command entry
F12=Cancel       F13=Repeat     F16=User options F24=More keys

```

Figure 68. Work with Programs Referencing Data Area Display

On this display you can edit, delete, compile, display object information, or search the programs listed.

In this example, however, we will exit without working with either program.

4. Press F3 (Exit) to return to the Application Dictionary Services/400 main menu.

Finding Programs, Service Programs, or Modules That Reference a File

The following example shows you how to find programs, service programs or modules that refer to a file:

1. From the Application Dictionary Services/400 menu, type 2 (Work with files) and press the Enter key.

The Subset Files to Work With display appears.

2. Specify the subsetting options needed, and press the Enter key to proceed to the Work with Files display.
3. Type 10 (Work with programs referencing file) beside the file for which you want the information (PDF in this case), and press the Enter key.

The Work with Programs Referencing File display appears, listing all programs, service programs, or modules that refer to PDF.

```

                                Work with Programs Referencing File
Dictionary . . . . . : ADLIB01DT      Attribute . . . . . : PF-DTA
File . . . . . : PDF                Library . . . . . : AP01.MST
Text . . . . . : Product File

Type options, press Enter.
 2=Edit          4=Delete          14=Compile          15=Scan RPG source
18=Display object information      25=Find string

Opt Program      Library  Type      Attribute  Use  Text
--  APDFI1       AP01.MST *PGM      RPG        01  Product data inquiry
--  CBLC         AP01.MST *MODULE   CBLLE      01  COBOL CALL C
--  CHGXXF       AP01.MST *SRVPGM   *          08
--  RPT01        AP01.MST *MODULE   RPGLE      01  RPGCALL CLP TEST
--  YEH1         AP01.MST *SRVPGM   *          08  Call yehcalling
--  YEH2         AP01.MST *SRVPGM   *          08  call yehcalling

                                Bottom
F3=Exit          F4=Prompt      F5=Refresh        F10=Command entry
F12=Cancel       F13=Repeat     F16=User options  F24=More keys

```

Figure 69. Work with Programs Referencing File Display. This example shows the programs, service programs, and modules that refer to a file.

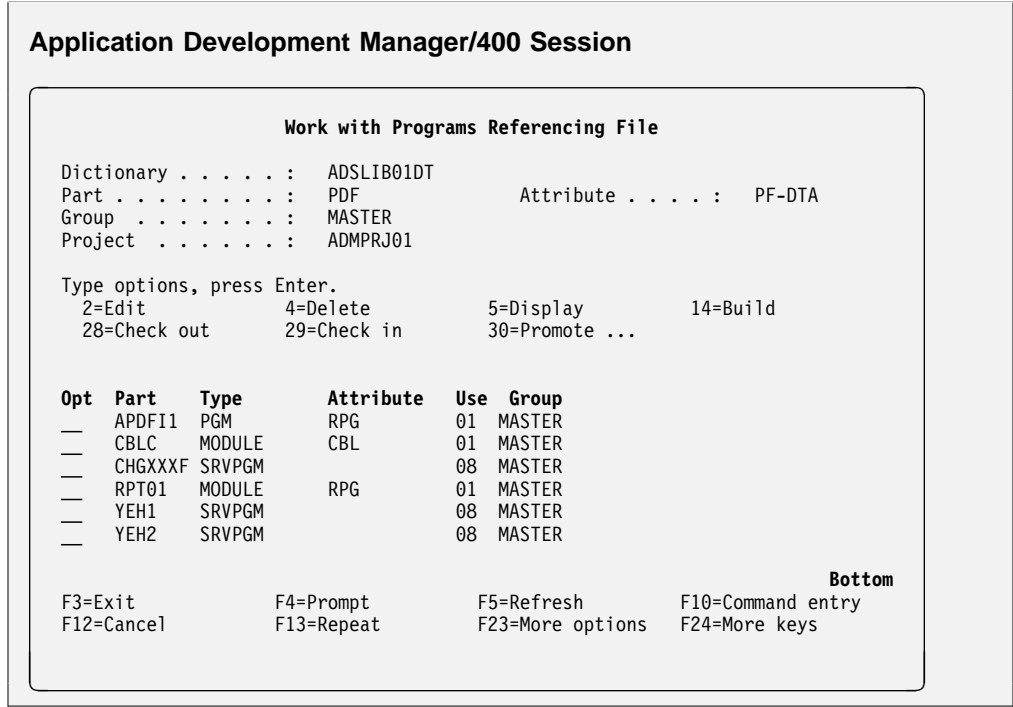


Figure 70. Work with Programs Referencing File Display in an Application Development Manager/400 Session. This example shows the programs, service programs, and modules that refer to a file.

The Use prompt provides information about how the file is used by the program or object.

Table 2 shows you the possible values for the Use field, and the types of files that can be used with each value. (An x indicates the file usages for each usage type.)

Table 2 (Page 1 of 2). The USE values that define how files are used

Use	Input file	Output file	Update file	Unspeci- fied file	External data structure ¹
01	x				
02		x			
03	x	x			
04			x		
05	x		x		
06		x	x		
07	x	x	x		
08				x	
09	x			x	
10		x		x	
11	x	x		x	
12			x	x	
13	x		x	x	
14		x	x	x	

Table 2 (Page 2 of 2). The USE values that define how files are used

Use	Input file	Output file	Update file	Unspecified file	External data structure ¹
15	x	x	x	x	
16					x

Note:

1. This usage applies to RPG/400 programs only.

Finding Files

You can find files if you know a:

- Physical file they are related to
- Logical file they are related to

Finding Files Related to a Physical File

A physical file often forms the basis for the definition of one or more logical files. The following example shows you how to determine which files are related to a given physical file.

1. From the Application Dictionary Services/400 menu, type 2 (Work with files), and press the Enter key.

The Subset Files to Work With display appears.

```

                                Subset Files to Work With
Dictionary . . . . . : ADSLIB01DT
Type choices, press Enter.
File . . . . . PDF          *ALL, name, generic*
Library . . . . . *ALL      *ALL, name, generic*
Attribute . . . . . *ALL      *ALL, attribute
                                F4 for list
Text:
Search words . . . . . _____
                                _____
Search condition . . . . . 1 1=Or, 2=And

F3=Exit      F4=Prompt      F5=Refresh      F12=Cancel
    
```

Figure 71. Subset Files to Work With Display

Application Development Manager/400 Session

```

Subset Files to Work With

Dictionary . . . . . : ADSLIB01DT
Project . . . . . : ADMPRJ01

Type choices, press Enter.

Part . . . . . PDF          *ALL, name, generic*
Attribute . . . . . *ALL      *ALL, attribute
                                F4 for list

Text:
  Search words . . . . . _____
                                _____

  Search condition . . . . . 1      1=Or, 2=And

F3=Exit      F4=Prompt      F5=Refresh      F12=Cancel
  
```

Figure 72. Subset Files to Work With Display in an Application Development Manager/400 Session

2. Fill in the fields as follows, and press the Enter key:
 - a. In the *File* or *Part* prompt, type the name of the file you want to work with (PDF in this case).
 - b. In the other prompts, type *ALL.

The Work with Files display appears.

```

Work with Files

Dictionary . . . . . : ADSLIB01DT
Position to . . . . . _____ Starting characters of file

Type options, press Enter.
  2=Edit      3=Copy      4=Delete      5=Display
  6=Design    7=Rename      9=Work with related files ...

Opt File      Library  Attribute  Text
9  PDF        AP01.MST  PF-DTA    Product File

Command Bottom
====> _____
F3=Exit      F4=Prompt      F5=Refresh      F6=Create
F9=Retrieve   F12=Cancel     F23=More options F24=More keys
This is a subsetted list.
  
```

Figure 73. Work with Files Display

Application Development Manager/400 Session

```

                                Work with Files
Dictionary . . . . . : ADSLIB01DT
Position to . . . . . _____ Starting characters of part

Type options, press Enter.
  2=Edit          3=Copy          4=Delete          5=Display
 13=Change       14=Build         28=Check out     29=Check in ...

Opt Part      Attribute  Group
  9_  PDF      PF-DTA    MASTER

Command _____ Bottom
===>
F3=Exit          F4=Prompt       F5=Refresh       F6=Create
F9=Retrieve      F12=Cancel      F23=More options F24=More keys
This is a subsetting list.
```

Figure 74. Work with Files Display in an Application Development Manager/400 Session



Tip:

In an Application Development Manager/400 session, AppDict Services/400 searches for the parts using the search path specified on the Start AppDict Services/400 (STRADS) display. Only the first occurrence of each file in the search path will be listed on the display.

3. Type 9 (Work with related files) beside the physical file for which you want to see the related logical files (PDF in this case), and press the Enter key.

Note: This option is supported only for files with attributes PF-DTA, LF, LF-JOIN, PF-SQLTBL, LF-SQLINX and LF-SQLVIW.

The Work with Files Related to Physical File display appears, listing all logical files related to PDF.

```

Work with Files Related to Physical File
Dictionary . . . . . : ADSLIB01DT
File . . . . . : PDF
Text . . . . . : Product File
Library . . . . . : AP01.MST

Type options, press Enter.
  2=Edit          4=Delete          5=Display
 10=Work with programs referencing file  12=Work with record formats ...

Opt File      Library  Attribute  Text
—  PDFL      AP01.MST  LF        PRODUCT FILE
—  TXFLA     AP01.MST  LF-JOIN   ORDER TXF FILE (JOIN LF)

F3=Exit          F4=Prompt      F5=Refresh    F10=Command entry
F12=Cancel       F13=Repeat     F23=More options  F24=More keys
Bottom

```

Figure 75. Work with Files Related to Physical File Display. This example shows the logical files that are related to a physical file.

```

Application Development Manager/400 Session

Work with Files Related to Physical File
Dictionary . . . . . : ADSLIB01DT   Part . . . . . : PDF
Group . . . . . : MASTER
Project . . . . . : ADMPRJ01

Type options, press Enter.
  2=Edit          4=Delete          5=Display          14=Build
 28=Check out    29=Check in      30=Promote ...

Opt Part      Attribute  Group
—  PDFL      LF        MASTER
—  TXFLA     LF-JOIN   MASTER

F3=Exit          F4=Prompt      F5=Refresh    F10=Command entry
F12=Cancel       F16=User options  F23=More options  F24=More keys
Bottom

```

Figure 76. Work with Files Related to Physical File Display in an Application Development Manager/400 Session. This example shows the logical files that are related to a physical file.

Finding Files Related to a Logical File

The following example shows you how to determine which files are related to a specific logical file.

1. From the Application Dictionary Services/400 menu, type 2 (Work with files), and press the Enter key.

The Subset Files to Work With display appears.

2. Specify the subsetting options needed, and press the Enter key to proceed to the Work with Files display.

3. Type 9 (Work with related files) beside the name of the logical file for which you want to see the related files (PDFL in this case), and press the Enter key.

The Work with Files Related to Logical File display appears, listing all files related to PDFL. (In this case there is only one, PDF.)

```

Work with Files Related to Logical File

Dictionary . . . . . : ADSLIB01DT
File . . . . . : PDFL           Library . . . . . : AP01.MST
Text . . . . . : PRODUCT FILE

Type options, press Enter.
  2=Edit           5=Display           10=Work with programs referencing file
 12=Work with record formats      14=Compile ...

Opt  File      Library  Attribute  Text
  _   PDF       AP01.MST  PF-DTA     Product File

Bottom
F3=Exit           F4=Prompt     F5=Refresh   F10=Command entry
F12=Cancel        F13=Repeat    F23=More options  F24=More keys
  
```

Figure 77. Work with Files Related to Logical File Display. This example shows the physical file that is related to a logical file.

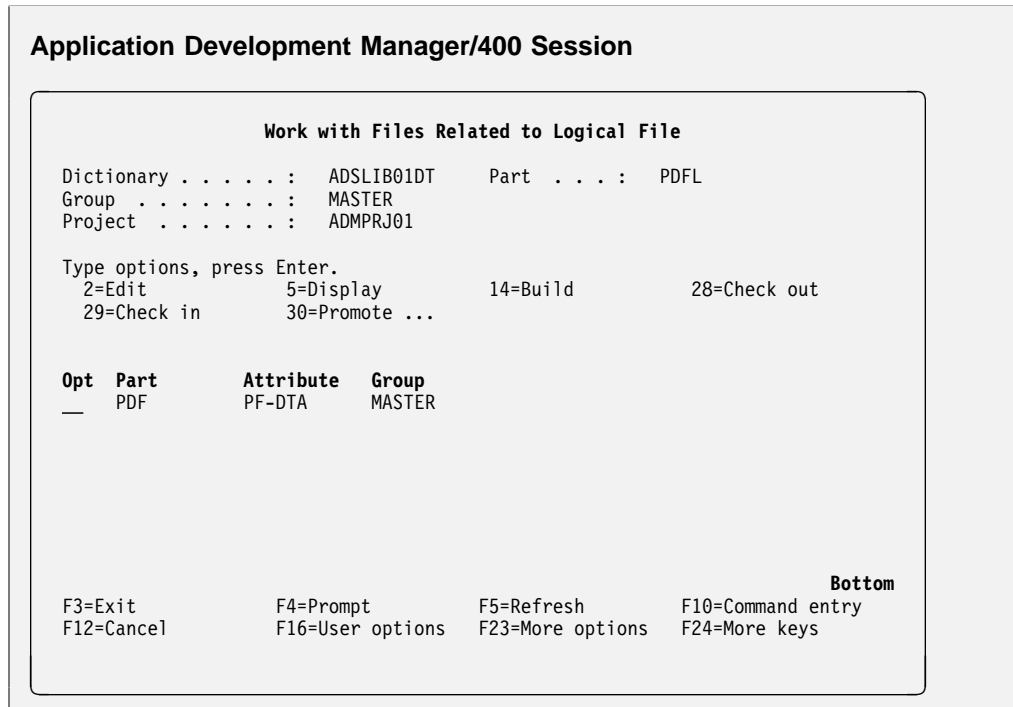


Figure 78. Work with Files Related to Logical File Display in an Application Development Manager/400 Session. This example shows the physical file that is related to a logical file.

Finding Files Used By a Program, Service Program, or Module

To determine what files are used by a program, service program, or module:

1. From the Application Dictionary Services/400 menu, type 3 (Work with programs and modules) to get to the Subset Programs and Modules to Work With display.
2. Specify the subsetting options needed, and press the Enter key to proceed to the Work with Programs and Modules display.
3. Type 10 (Work with files used) beside a program (YEH1 in this case) to display a list of files used in it, and press the Enter key.

Note: This option only supports BASIC, CL, COBOL, PL/I and RPG programs and OCL36 procedures, and the ILE equivalents of these program and procedure types. (One exception applies: file references are not available for CLE modules.)

The Work with Files Used display appears, listing the file used in this program.

```

Work with Files Used
Dictionary . . . . . : ADSLIB01DT      Library . . . . . ; AP01.MST
Object . . . . . : YEH1                Type . . . . . : *PGM
Text . . . . . : Call yehcalling

Type options, press Enter.
  2=Edit          4=Delete          5=Display          6=Design
 10=Work with programs referencing file      14=Compile ...

Opt File      Record      Use Library      Attribute      Record text
—   PDF                08 AP01.MST      PF-DTA

Bottom
F3=Exit          F4=Prompt        F5=Refresh        F10=Command entry
F12=Cancel       F16=User options F23=More options  F24=More keys

```

Figure 79. Work with Files Used Display

```

Application Development Manager/400 Session

Work with Files Used
Dictionary . . . . . : ADSLIB01DT      Part . . . . . : YEH1
Group . . . . . : MASTER              Type . . . . . : PGM
Project . . . . . : ADMPRJ01

Type options, press Enter.
  2=Edit          4=Delete          5=Display          29=Check in ...
 13=Change       14=Build         28=Check out

Opt Part      Record      Attribute      Use Group
—   PDF                PF-DTA        08 MASTER

Bottom
F3=Exit          F4=Prompt        F5=Refresh        F10=Command entry
F12=Cancel       F16=User options F23=More options  F24=More keys

```

Figure 80. Work with Files Used Display in an Application Development Manager/400 Session

Finding Information About Record Formats Defined in Files

To find information about a record format:

1. From the Application Dictionary Services/400 menu, type 2 (Work with files) and press the Enter key.

The Subset Files to Work With display appears.

2. Specify the subsetting options needed, and press the Enter key to proceed to the Work with Files display.

3. Type 12 (Work with record formats) beside the name of the file for which you want the record format information (PDF in this case), and press the Enter key.

Note: You cannot type this option when the file attribute is ICFF, SAVF, PF-SRC, DDMF, PF-SQLTBL, LF-SQLVIW, or LF-SQLINX.

The Work with Record Formats display appears.

Work with Record Formats

Dictionary : ADLIB01DT
File : PDF Attribute : PF-DTA
Library : AP01.MST
Text : Product File

Type options, press Enter.
11=Work with fields in record format

Opt	Record	Field	Length	Level ID	Text
<u>11</u>	PDFREC	5	30	3A0E1A43636E6	

Bottom

F3=Exit F4=Prompt F5=Refresh F10=Command entry
F12=Cancel F13=Repeat F24=More Keys

Figure 81. Work with Record Formats Display. This example shows the record formats being selected.

Application Development Manager/400 Session

```

                                Work with Record Formats

Dictionary . . . . . : ADSLIB01DT
Part . . . . . : PDF           Attribute . . . . . : PF-DTA
Group . . . . . : MASTER
Project . . . . . : ADMPRJ01

Type options, press Enter.
  11=Work with fields in record format

Opt Record      Field Length Level ID      Text
11 PDFREC          5      30  3A0E1A43636E6

                                Bottom

F3=Exit          F4=Prompt      F5=Refresh      F10=Command entry
F12=Cancel       F16=User options F21=Print list  F24=More keys
  
```

Figure 82. Work with Record Formats Display in an Application Development Manager/400 Session. This example shows the record formats being selected.

4. Type 11 (Work with fields in record format) beside the name of the record whose fields you want to see (PDFREC in this case), and press the Enter key. The Work with Fields in Record Format display appears.

```

                                Work with Fields in Record Format

Dictionary . . . . . : ADSLIB01DT
File . . . . . : PDF           Attribute . . . . . : PF-DTA
Library . . . . . : AP01.MST  Record . . . . . : PDFREC
Text . . . . . : Product File

Type options, press Enter.
  5=Display          8=Work with impacted files
  20=Work with objects to recreate

Opt Field      Input Output Type Key Length Decimal
offset offset
5_ PDNAME        4      4      A    1     20
— PDNO          1      1      A    1      3
— PRICE        24     24     S    3      0
— QTY          27     27     P    3      0
— UNIT         29     29     A    2
  
```

Figure 83. Work with Fields in Record Format Display. This example shows the fields available.

Application Development Manager/400 Session

```
Work with Fields in Record Format

Dictionary . . . . . : ADSLIB01DT      Part . . . . . : PDF
Attribute . . . . . : PF-DTA          Record . . . . . : PDFREC
Group . . . . . : MASTER
Project . . . . . : ADMPRJ01

Type options, press Enter.
  5=Display          8=Work with impacted files

Opt  Field      Input  Output  Type  Key  Length  Decimal
  5_ PDNAME      4      4      A     1    20
  — PDNO        1      1      A     1     3
  — PRICE      24     24     S     1     3     0
  — QTY        27     27     P     1     3     0
  — UNIT       29     29     A     1     2

F3=Exit          F4=Prompt      F5=Refresh     F6=Recreate parts
F11=Display text F12=Cancel     F24=More keys
```

Figure 84. Work with Fields in Record Format Display in an Application Development Manager/400 Session. This example shows the fields available.

This display permits you to look at the record format layout and also provides you with information about the key fields used in the record format. It is sorted by field name.

5. If you want to see the fields sorted by **input offset** (which is the character position that indicates the start of the field starting from the beginning of the record), press F15 (Sort input offset).

- a. Type 5 (Display) beside the name of the field containing the information you want to see (PDNAME in this case), and press the Enter key.

The Display Field Information display appears.

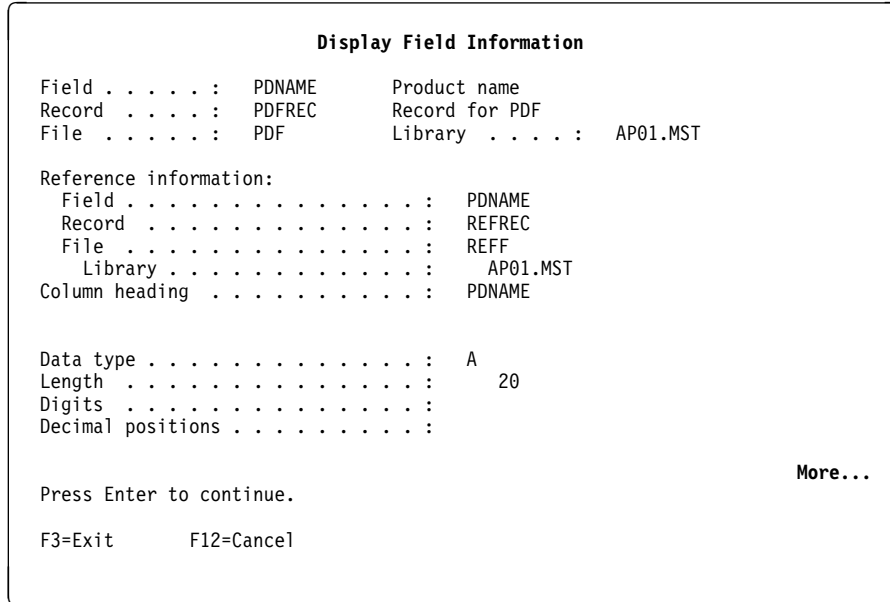


Figure 85. Display Field Information Display

6. Press the Enter key to go to the Work with Fields in Record Format display.
7. Repeat steps 4 through 6 to display the fields in other record formats.

Finding Data Areas Used By Programs, Service Programs, and Modules

To determine what data areas are used by a program, service program, or module:

1. From the Application Dictionary Services/400 menu, type 3 (Work with programs and modules) to get to the Subset Programs and Modules to Work With display.
2. Specify the subsetting options needed, and press the Enter key to proceed to the Work with Programs and Modules display.
3. Type 12 (Work with data areas used) beside a program, service program, or module (YEH1 in this case) to display a list of data areas used in it, and press the Enter key.

Note: This option only supports programs with the attribute RPG or CLP, or either of the ILE equivalents (RPGLE or CLLE). Although this option only pertains to RPGLE or CLE modules, it can also be used with CLE and CBLLE programs or service programs, since these objects may contain RPGLE or CLLE modules that reference data areas.

The Work with Data Areas Used display appears, listing the data areas used in this program.

```

Work with Data Areas Used
Dictionary . . . . . : ADSLIB01DT      Library . . . . . : AP01.MST
Program . . . . . : YEH1              Type . . . . . : *PGM
Text . . . . . : Call yehcalling

Type options, press Enter.
  4=Delete          5=Display
  10=Work with programs referencing data area ...

Opt Data area Library Text
  _  DTAARA      *LIBL      (Object not documented.)

Bottom
F3=Exit          F4=Prompt      F5=Refresh      F10=Command entry
F12=Cancel       F16=User options F23=More options F24=More keys

```

Figure 86. Work with Data Areas Used Display

```

Application Development Manager/400 Session

Work with Data Areas Used
Dictionary . . . . . : ADSLIB01DT      Part . . . . . : YEH1
Group . . . . . : MASTER              Type . . . . . : PGM
Project . . . . . : ADMPRJ01

Type options, press Enter.
  4=Delete          5=Display          13=Change          28=Check out
  29=Check in      30=Promote ...

Opt Part Group Text
  _  DTAARA      (Object not documented.)

Bottom
F3=Exit          F4=Prompt      F5=Refresh      F10=Command entry
F12=Cancel       F16=User options F23=More options F24=More keys

```

Figure 87. Work with Data Areas Used Display in an Application Development Manager/400 Session

4. If you want to display the programs, service programs, or modules that refer to a specified data area, type 10 (Work with programs referencing data area) beside the data area you want in the Work with Data Areas Used display, and press the Enter key.

Finding Objects By the Last Changed Source Date

The AppDict Services/400 feature allows you to find program and module objects based on the date that their source was last changed. If you specify a range of time, AppDict Services/400 will display a list of all of the documented program and module objects with a last changed source date that falls in that range of time.

To list program and module objects by their last changed source date:

1. From the Application Dictionary Services/400 menu, type 5 (Work with objects by last changed source), and press the Enter key.

The Subset Objects by Last Changed Source List display appears.

2. Fill in the prompts as follows, and press the Enter key:

- a. In the *From date* prompt, type the starting date in the range of time you are going to search. Type the date in the system format.
- b. In the *To date* prompt, type the ending date in the time range.
- c. In the *Object* or *Library* prompt (or *Part* prompt if you are working in an Application Development Manager/400 session), *Type*, *Attribute*, and *Text* prompts, type the rest of the subsetting criteria that you require, and press the Enter key.

The Work with Objects by Last Changed Source display appears, listing the program and module objects with a last changed source date falling in the range you specified.

```

                                Work with Objects by Last Changed Source
Dictionary . . . . . :  ADLIB01DT
Position to . . . . . :  _____ Date of object

Type options, press Enter.
12=Work with

Opt Object      Library   Type      Attribute  Date      Time
8  YEHI         AP01.MST   *PGM      CLP         12/01/95  14:58:02

Command _____ Bottom
===>
F3=Exit          F4=Prompt      F5=Refresh     F9=Retrieve
F12=Cancel       F13=Repeat     F17=Subset     F24=More keys
This is a subsetting list.
```

Figure 88. Work with Objects by Last Changed Source Display. This example shows the program object that falls within the time period specified.

Application Development Manager/400 Session

Work with Objects by Last Changed Source

Dictionary : ADSLIB01DT
Position to _____ Date of part

Type options, press Enter.
12=Work with

Opt	Part	Date	Time	Group
<u>8</u>	YEH1	12/01/95	14:50:02	MASTER

Bottom

Command
===>

F3=Exit	F4=Prompt	F5=Refresh	F9=Retrieve
F11=Display type/attribute	F12=Cancel	F24=More keys	

This is a subsetted list.

Figure 89. Work with Objects by Last Changed Source Display in an Application Development Manager/400 Session. This example shows the program object that falls within the time period selected.

3. You can specify option 12 (Work with) to do any of the functions associated with that object.

Chapter 9. Maintaining the Application Dictionary Services/400 Tool

This chapter describes the periodic maintenance tasks you can perform on your AppDict Services/400 system:

- Checking the journals to see that the dictionaries are automatically refreshing properly
- Backing up the AppDict Services/400 system (QDMT library)

Ensuring Your Dictionaries Are Automatically Refreshing

The function of automatic refreshing is very similar to rebuilding a dictionary. AppDict Services/400 automatically refreshes the dictionaries as you work with them. However, this process can fail when certain parts of the AS/400 system are not set up correctly. For example, automatic refreshing fails if an incorrect value is set for the system value QAUDLVL.

This section describes:

- More technical information about automatic refreshing for the security officer
- How to determine whether automatic refreshing is working
- How to rebuild dictionaries if automatic refreshing fails

More Technical Information About Automatic Refreshing

The following description, which is more technical, is meant for the security officer. General users need not read this section.

When the AppDict Services/400 feature is first started after installation, it starts the subsystem QDNNOTIFY, with the QDNDCTSYNC job running in it to take care of system events. The system value QAUDLVL (security auditing level) needs to be set to *SECURITY, *CREATE, *DELETE, *OBJMGT, and *SAVRST. If these settings do not already exist, they are automatically added, and a message is sent to the message queue of QSYSOPR telling it that the system value QAUDLVL has been changed. These settings must be in place; they will be repeatedly added to QAUDLVL if you try to remove them.

A journal, QAUDJRN, is created in library QSYS, and a journal receiver, QAUDRCxxxx, is attached to this journal. If the QAUDJRN journal exists, the current attached journal receiver name is used. The string xxxx of the attached journal receiver increases from 0000 to 9999. There can be up to 9999 journals, and each journal can have up to 99 999 journal entries. The entry number of a journal receiver may exceed 99 999. Every time this happens, a new journal receiver is created by using JRNRCV(*GEN) parameter on the CHGJRN command.

If the creation of the journal receiver fails, an Unexpected error occurred message appears. This could happen because either the name already exists or the attached receiver is damaged. The security officer must then delete this journal receiver. Before deleting it, the security officer may want to back up this journal receiver.

Determining Whether or Not Automatic Refreshing Is Working

However, if certain things on your AS/400 system change, the data in your dictionaries may not be automatically refreshing. As a result, your dictionaries may not be synchronized. For example, the following can prevent automatic refreshing from working:

- Changing the system value QAUDLVL

Note: It is recommended that you do not change the QAUDLVL value during an AppDict Services/400 session.

- Creating a new journal receiver to replace the current journal receiver (for example, when the sequence number reaches 99 999 and the current journal becomes full). During the time you are creating the new journal receiver, other activities may be occurring on the AppDict Services/400 feature which will not be documented in the journal.

If automatic refreshing is not working, you will need to rebuild the dictionary so that it is synchronized with the AppDict Services/400 feature. For information about synchronizing a dictionary, see “Rebuilding a Dictionary” on page 121.

There are two ways to determine whether all of the entries in a journal have been processed, and that automatic refreshing is working. These are described in the next two sections.

First Method

The first method is the easier one. Issue the WRKACTJOB (Work with Active Jobs) command to see the status of job QDNDCTSYNC under the QDNNOTIFY subsystem. The status may be one of the following:

Status	Description
RUN (Running):	The job is manipulating the entries in the journal and in your dictionaries.
TIMW (Time Wait):	The job has processed all entries, and is waiting for a new entry to be logged in the journal.
LCKW (Lock Wait):	The job is waiting for a lock on an object to be released.

Press F5 (Refresh) periodically, to update the status of the jobs because that status varies from time to time.

Second Method

Note: To use this method, you must have authority to the QAUDJRN journal.

The second method is to display the contents of journal QAUDJRN in library QSYS, by using the DSPJRN JRN(QAUDJRN) command. In the list, search for the last entry whose code is U and whose type is CM. All entries up to this one were processed successfully; all entries following it are still being processed. If this entry is the last one in the list, this means that no new system events have been logged in the journal.

Because the number of entries in the journal could be large, you may want to display the contents starting from a particular journal entry. If so, use the DSPJRN JRN(QAUDJRN) FROMENT(xxxxx) command, where xxxxx is the starting sequence number from 0 to 99 999 (for example, 35 000).

Rebuilding a Dictionary

The information in your dictionary is usually up-to-date because the AppDict Services/400 automatic-refresh function is always running. Only rebuild your dictionary if you have discovered that the information in the dictionary is not current. For more information about determining whether or not automatic refreshing is working, refer to "Determining Whether or Not Automatic Refreshing Is Working" on page 120.

Note: Only a dictionary whose status is Not in use or Not complete can be rebuilt.

To rebuild a dictionary:

1. On the Work with Dictionaries display type 9 (Rebuild) beside the dictionary name you choose, and press the Enter key.

The Submit to Rebuild Dictionary display appears.

```

                Submit to Rebuild Dictionary
Dictionary . . . . . : ADSLIB01DT
Type choice, press Enter.
  Hold job . . . . .  N                Y=Yes, N=No

F3=Exit          F5=Refresh          F12=Cancel
```

Figure 90. Submit to Rebuild Dictionary. This example shows the dictionary rebuild job being confirmed.

2. Press the Enter key to submit a batch job to rebuild the dictionary.

You are returned to Work with Dictionaries display. A message appears at the bottom of the display to tell you that the dictionary rebuilding job has been submitted.

After you submit the rebuild job, the status of the dictionary will be Processing or Not in use. The former message means that you do not need to hold the job, and that the job is either processing or waiting for processing. The latter message means that you need to hold the job and then release it before the job can begin processing.



Tip:

If you have specified Y in the *Hold job* prompt, the dictionary rebuilding job will not be processed until you release this held job. Before you release it, you must make sure that the dictionary is not being used by other users. (You can use the WRKAPPDCT command to verify this. For more information about using the WRKAPPDCT command, refer to Appendix A.) If you release the held job while the dictionary is in the In use status, an unexpected result will occur.

To release the held job:

1. Type the WRKSBMJOB command on the command line to call the Work with Submitted Jobs display.
2. Select option 6 (Release) beside the rebuilding job, and press the Enter key.

Backing Up the Application Dictionary Services/400 System

All of the files that are important to the AppDict Services/400 feature and your development projects are stored in the QDMT library. To allow you to recover your AppDict Services/400 system in the event that a hardware failure occurs or files become corrupted, you should periodically back up the QDMT library.

To back up the QDMT product library, follow these steps:

1. To display the QDMT and the QDNNOTIFY subsystems, type:

```
WRKACTJOB SBS(QDMT QDNNOTIFY)
```

The Work with Active Jobs display appears.

2. If these subsystems are not active, use the Save Library (SAVLIB) command to save the QDMT library. Otherwise, continue with the rest of the steps.
3. The status of the QDNNOTIFY job can be either RUN or TIMW. Check the status of the QDNNOTIFY subsystem, and end it based on the following conditions:

Note: Do not terminate the QDNDCTSYNC job. The QDNDCTSYNC job should always be active, as this job is used to auto-refresh all the documented dictionaries.

- If its status is RUN, then there is at least one auto-refresh activity going on in the subsystem. Wait until this activity ends before you end this subsystem.
- If its status is TIMW, then there is no active auto-refresh activity going on. You can end the subsystem immediately.

Note: When the QDNNOTIFY subsystem ends, all the updates to the dictionaries will be logged in the journal. These updates will be refreshed when the AppDict Services/400 feature is restarted.

4. Check the status of the QDMT subsystem.

It is possible to have more than one job appear in this subsystem: for example, SERVER1, SERVER2, SERVER3, SERVER4, and SERVER5. In some cases there may be multiple SERVERs that are all used to update and document dictionaries.

5. End the QDMT subsystem based on the following conditions:

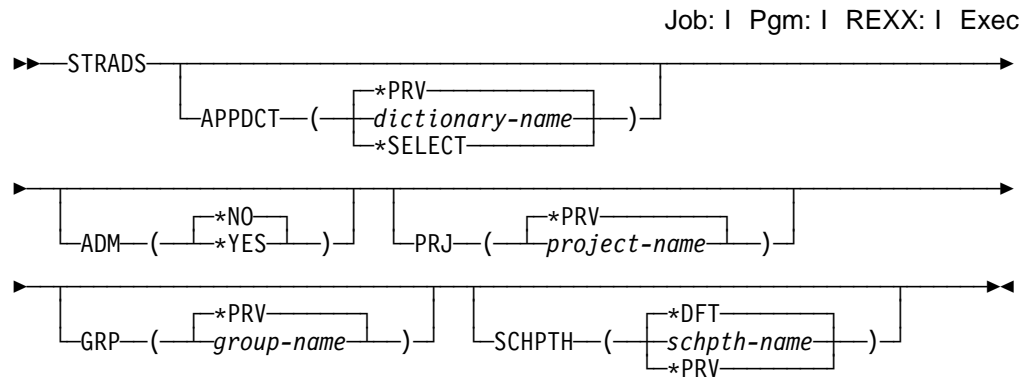
Note: The SERVER1 job should always run.

- a. It is important to check the status of the SERVER2, SERVER3, SERVER4, and SERVER5 jobs; however, they may not appear when you display active jobs. They may have a status of RUN, MSGW, or LCKW.
 - b. Jobs with LCKW status will eventually end themselves.
 - c. If there are no jobs besides the SERVER2, SERVER3, SERVER4, and SERVER5 jobs with a RUN status, end the QDMT subsystem.
6. When the QDMT and QDNNOTIFY subsystems end, you may use the SAVLIB command to back up the QDMT library.

Appendix A. Control Language Commands in Application Dictionary Services/400

This appendix lists the control language (CL) commands that are specific to the Application Dictionary Services/400 (AppDict Services/400) feature. Each of the command parameters is followed by a description of its use.

STRADS (Start AppDict Services/400) Command



Purpose

The STRADS command allows you to start the AppDict Services/400 feature. Type STRADS on the command line, and then press F4 (Prompt). The Application Dictionary Services/400 display appears.

Required Parameters

APPDCT

Specify the name of the dictionary that you want to use.

The possible values are:

***PRV:** Specify that the dictionary used in the previous session will be used in this session.

dictionary-name: Type the name of a dictionary that you want to use. The dictionary name must be an existing library on the system. If the specified library is not a dictionary in AppDict Services/400, the library will be treated as a new library.

***SELECT:** Specify that a list of dictionaries be displayed from which you can select a dictionary.

Optional Parameters

ADM

Specify whether or not you want to use an Application Development Manager/400 session.

The possible values are:

***NO:** Specify that you want to work directly in an AppDict Services/400 session.

***YES:** Specify that you want to work in AppDict Services/400 from within an Application Development Manager/400 session.

PRJ

Specify the name of a project with which you want to work.

The possible values are:

***PRV:** Specify that the project used in the previous session will be used in this session.

project-name: Type the name of the project that you want to use.

GRP

Specify a group name to use as the initial group of the search path. Search the SCHPTH part beginning from the group specified, and on down. If the dictionary is not found there, search up from this group in the project hierarchy. AppDict Services/400 will also search for the SCHPTH part, itself, starting in this group and working up the project hierarchy.

The group specified for this parameter will be the working group when you are in an Application Development Manager/400 session.

The possible values are:

***PRV:** Specify that the group used in the previous session will be used in this session.

group-name: Type the name of the group that you want to use.

SCHPTH

Specify a search-path part name that will be used to produce a subsetted list of parts and to build parts.

The possible values are:

***DFT:** Specify that the search path will use QDFT as the default value. If that part is not found, the default project hierarchy will be used.

schpth-name: Specify a specific search-path name that you want to use.

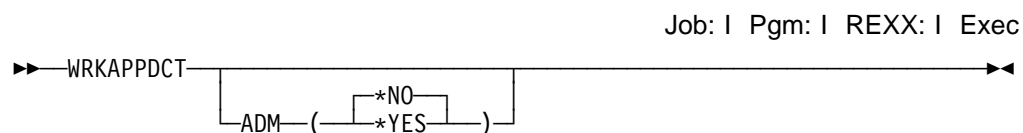
***PRV:** Specify that the search path used in the previous session will be used in this session.

Example

```
STRADS APPDCT(ADSLIB01DT) ADM(*YES) PRJ(ADMPRJ01) GRP(MASTER) SCHPTH(*DFT)
```

This command allows you to start the AppDict Services/400 feature in an Application Development Manager/400 session specifying application dictionary ADSLIB01DT, project ADMPRJ01, group MASTER, and search path QDFT.

WRKAPPDCT (Work with Dictionaries) Command



Purpose

The WRKAPPDCT command allows you to bypass the Application Dictionary/400 menu when starting the AppDict Services/400 feature. Type WRKAPPDCT on the command line, and then press F4 (Prompt). The Work with Dictionaries display will appear.

Optional Parameters

ADM

Specify whether or not you want to use an Application Development Manager/400 session.

The possible values are:

***NO:** Specify that you want to work directly in AppDict Services/400.

***YES:** Specify that you want to work from within an Application Development Manager/400 session.

Example

```
WRKAPPDCT ADM(*NO)
```

This command allows you to bypass the prompted Start AppDict Services/400 (STRADS) display and go directly to the Work with Dictionaries display. This command allows you to access AppDict Services/400 directly (not from within an Application Development Manager/400 session).

Appendix B. System/36 and System/38 Restrictions

Application Dictionary Services/400 (AppDict Services/400) is not designed for System/36 and System/38 objects. Several of the AppDict Services/400 functions do not work with System/36 and System/38 objects.

In the Work with Fields display, all options work, but the information may not be complete because of the System/36 or System/38 command restrictions. For example, you can display the field references, but if you want to recreate objects, some System/36 and System/38 programs may be lost because the DSPPGMREF command does not support System/36 and System/38 language.

In the Work with Files display, all options work, except that you cannot create System/36 and System/38 files by pressing F6 (Create). However, AppDict Services/400 will scan operator control language (OCL) in a library and try to extract relationships between files and programs using the OCL procedure information. Normally, if you select option 6 (Design) to design a PF-DTA file, and if the source member does not exist, the source member will be retrieved, and you can then design a physical file using the design tool. However, this function does not support System/36 or System/38 physical files: for these, when you select option 6 (Design), you will be placed in the Source Entry Utility (SEU).

In the Work with Programs and Modules display, most of the program reference information is not complete for System/36 or System/38 programs because of the DSPPGMREF command restrictions. You cannot create System/36 or System/38 programs by pressing F6 (Create).

Only the following System/36 or System/38 attributes are supported:

- DSPF36
- DSPF38
- PRTF38
- PF38
- LF38
- BAS36
- BAS38
- CBL36
- CBL38
- CLP38
- OCL36
- PLI38
- RPG36
- RPG38

Bibliography

This bibliography lists a variety of books that may be of use or interest to you as you work with the Application Dictionary Services/400 feature.

The Application Dictionary Services/400 library consists of the following publications:

- *ADTS/400: Application Dictionary Services/400 Self-Study Guide*, SC09-2086
- *ADTS/400: Application Dictionary Services/400 User's Guide*, SC09-2087

The Application Development ToolSet/400 library consists of the following publications:

- *ADTS/400: Advanced Printer Function*, SC09-1766
- *ADTS/400: Character Generator Utility*, SC09-1769
- *ADTS/400: Data File Utility*, SC09-1773
- *ADTS/400: File Compare and Merge Utility*, SC09-1772
- *ADTS/400: Interactive Source Debugger*, SC09-1897
- *ADTS/400: Programming Development Manager*, SC09-1771
- *ADTS/400: Report Layout Utility*, SC09-1767
- *ADTS/400: Screen Design Aid*, SC09-1768
- *ADTS/400: Source Entry Utility*, SC09-1774
- *Introducing ADTS/400 and AS/400 Server Access Programs*, GC09-2088
- *LPS: Application Development ToolSet for OS/400*, GC09-2089

The Application Development Manager/400 library contains the following publications:

- *ADTS/400: Application Development Manager/400 API Reference*, SC09-2180
- *ADTS/400: Application Development Manager/400 Introduction and Planning Guide*, GC09-1807
- *ADTS/400: Application Development Manager/400 User's Guide*, SC09-2133
- *ADTS/400: Application Development Manager/400 Self-Study Guide*, SC09-2138

The following publications in the AS/400 library may be of interest to you in relation to this feature:

- *Publications Reference*, SC41-4003
- *Backup and Recovery – Basic*, SC41-4304
- *CL Programming*, SC41-4721
- *CL Reference*, SC41-4722
- *Data Management*, SC41-4210
- *DB2 for OS/400 Database Programming*, SC41-4701
- *DB2 for OS/400 SQL Programming*, SC41-4611
- *DB2 for OS/400 SQL Reference*, SC41-4612
- *Experience RPG IV*, SC09-1938
- *ILE Application Development Example*, SC41-3602
- *ILE Concepts*, SC41-4606
- *Programming Reference Summary*, SX41-4720
- *Software Installation*, SC41-4120
- *System Operation*, SC41-4203
- *Work Management*, SC41-4306

Index

A

- about this book ix
- administration
 - automatic refreshing 119
 - backing up 122
 - dictionary, fixing 120
 - dictionary, rebuilding 121
 - system values and 119
- Application Development Manager/400
 - recreating parts 85
 - starting session in 12
 - supported functions 7
- Application Dictionary Services/400
 - application development 6
 - functions 6
 - installing 9
 - introduction 1
 - starting 10
 - in Application Development Manager/400 session 12
 - selecting a dictionary 15
 - stand-alone 10
 - using WRKAPPDCT command 14
 - subsystems 25
 - transferring old dictionaries 9
 - user environments
 - defaults, changing 19
 - description 4, 19
 - example, user-defined options 24
 - multiple sessions 24
 - substitution variables 20
 - subsystems 25
 - user-defined options, changing 20
 - user-defined options, creating 20
- application objects
 - descriptions 1
 - relationships 1
- attention key function 25
- authority, dictionary 30
- automatic refreshing 4, 119

B

- backing up QDMT library 122
- batch
 - creating objects in dictionary 84
 - recreating objects 84
 - removing documentation from a dictionary 34

C

- calling structure
 - displaying ILE 97
 - displaying OPM 91
 - introduction 90
- CHGCLS command 25
- CL (Control Language) commands
 - STRADS (Start AppDict Services/400) 12, 125
 - WRKAPPDCT (Work with Application Dictionaries) 127
 - WRKAPPDCT (Work with Dictionaries) 14
- commands
 - Change Class (CHGCLS) 25
 - CHGCLS (Change Class) 25
 - Copy File (CPYF) 81
 - CPYF (Copy File) 81
 - Create Library (CRTLIB) 28
 - Create Physical File (CRTPF) 56
 - Create Program (CRTPGM) command
 - ENTMOD parameter 40
 - EXPORT parameter 40
 - Create Service Program (CRTSRVPGM) command
 - ENTMOD parameter 40
 - EXPORT parameter 40
- creating 61
- CRTLIB (Create Library) 28
- CRTPF (Create Physical File) 56
- Delete Licensed Program (DLTLICPGM) 9
- Display Journal (DSPJRN) 120
- Display Message (DSPMSG) 29, 31
- DLTLICPGM (Delete Licensed Program) 9
- DSPJRN (Display Journal) 120
- DSPMSG (Display Message) 29, 31
- End Subsystem (ENDSBS) 25
- ENDSBS (End Subsystem) 25
- Go Licensed Program (GO LICPGM) 9
- GO LICPGM (Go Licensed Program) 9
- Start AppDict Services/400 (STRADS) 10, 125
- Start Debug (STRDBG) 16
- STRADS (Start AppDict Services/400) 10, 125
- STRDBG (Start Debug) 16
- user-defined options 20
- Work with Active Jobs (WRKACTJOB) 120, 122
- Work with Application Dictionaries (WRKAPPDCT) 127
- Work with Dictionaries (WRKAPPDCT) 14
- Work with Spooled Files (WRKSPLF) 90
- Work with Submitted Jobs (WRKSBMJOB) 122
- WRKACTJOB (Work with Active Jobs) 120, 122
- WRKAPPDCT (Work with Application Dictionaries) 127

commands (*continued*)
 WRKAPPDCT (Work with Dictionaries) 14
 WRKSBJOB (Work with Submitted Jobs) 122
 WRKSPLF (Work with Spooled Files) 90
 compile after editing 20
 compilers
 AS/400 100
 ILE 100
 compiling
 example of order 77
 order of 77
 convention used in this book x
 Copy File (CPYF) command 81
 copying data 81
 CPYF (Copy File) command 81
 Create Library (CRTLIB) command 28
 Create Physical File (CRTPF) command 56
 Create Program (CRTPGM) command
 ENTMOD parameter 40
 EXPORT parameter 40
 Create Service Program (CRTSRVPGM) command
 ENTMOD parameter 40
 EXPORT parameter 40
 CRTLIB (Create Library) command 28
 CRTPF (Create Physical File) command
 parameters, changing for 56
 using 56

D

data areas
 by last changed source date, finding 116
 programs referencing 100
 used by modules 114
 used by programs 114
 used by service programs 114
 data file utility (DFU)
 creating a temporary program 24
 running a temporary program 24
 data items, sharing in ILE 40
 DB2/400 Query Manager 59
 DDS
 See DDS (Data Description Specifications)
 DDS (Data Description Specifications)
 displaying source 55
 saving for physical file creation 55
 DDS source, saving 55
 debug session, starting parameter 16
 default
 environment settings, changing 19
 user-defined option file 22
 Delete Licensed Program (DLTLICPGM) command 9
 derived information 5
 DFU
 See data file utility (DFU)

dictionary
 automatic refreshing 119
 affecting functions 4
 creating objects and 6
 deleting objects and 4, 6
 description 4
 moving objects and 4
 renaming objects and 4
 batch creation job 29, 31
 batch documenting job 33
 contents 5
 creating 29, 30
 first 10
 general 28, 30
 in Application Development Manager/400
 session 30
 in stand-alone Application Dictionary
 Services/400 28
 definition 4
 deleting 36
 disk space usage 35
 disk space, recovering 35
 documentation addition batch job 34
 documentation removal batch job 35
 finding documented 36
 fixing 120
 group documentation, adding 33
 groups, adding 33
 groups, removing documentation of 34
 information stored in 5
 information, changing 1
 library documentation, adding 32
 library, removing documentation of 34
 multiple projects and 37
 naming 29, 30
 rebuilding 121
 rebuilding batch job 121
 reorganizing 35
 reorganizing batch job 36
 status
 description 28
 in use 28
 not complete 28
 not in use 28
 processing 28
 stored information types
 derived 5
 meta 6
 static 5
 transferring, previous release 9
 use 4
 verifying build 29, 31
 disk space analysis 35
 Display Journal (DSPJRN) command 120
 Display Message (DSPMSG) command 29, 31

displaying messages 29, 31
DLTLICPGM (Delete Licensed Program) command 9
DSPJRN (Display Journal) command 120
DSPMSG (Display Message) command 29, 31

E

editing source with SEU 80
ENDSBS command 25
ENTMOD parameter 40
examples
 using the STRADS command in Application Development Manager/400 session 126
 using the WRKAPPDCT command 127
EXPORT parameter 40

F

field reference files
 creating
 introduction 47
 physical file for 47
 marking 65
 physical file creation for 47
 physical files for
 DDS, saving for 55
 field definitions 50
 file-level keyword definition 49
 key field definitions 53
 record-level keyword definition 50
 removing the mark from 68
field reference hierarchy
 determining 69
 example 75
 levels, described 73
field reference level description 74
fields
 affects of changing
 field reference hierarchy 69
 for files 75, 78
 for programs 78
 for service programs 78
 impacted files 69
 impacted programs 69
 introduction 69
 recreating affected objects 81
 changing 80
 defining 50
 definitions for physical files 50
file-level keywords, defining 49
files
 access types, specifying 57
 affects of changing
 introduction 69
 by last changed source date, finding 116
 creating
 access types, specifying 57

files (*continued*)
 creating (*continued*)
 logical files 56
 types 46
 field reference file
 physical file for 47
 physical file, creating for 47
 field reference files
 introduction 47
 marking 65
 removing the mark from 68
 finding related to logical 108
 finding related to physical 104
 impacted 75
 physical file
 DDS source, saving for 55
 field definitions for 50
 file-level keyword definition for 49
 key field definitions for 53
 record-level keyword definition for 50
 record formats used in, finding 111
 reference level description 77
SQL
 batch table file creation job for 59
 index files based on physical files for 61
 index files based on table files for 59
 index files batch creation job for 61
 introduction 57
 table files based on field reference files 57
 view files based on physical files 61
usages 103
usages, description 103
used by modules 109
used by programs 109
used by service programs 109
find string
 and scanning RPG source 87
 in source members 88
 using 88
 using PDM 88
functions
 Application Development Manager/400 supported 7
 stand-alone 6

G

Go Licensed Program (GO LICPGM) command 9
GO LICPGM (Go Licensed Program) command 9
groups
 documenting 33
 removing documentation of 34

H

held job, releasing 122

help, getting 17

I

ILE

See Integrated Language Environment (ILE)

impact analysis

default options for recreating affected objects 81

field change on files 75

field reference hierarchy

example 75

levels 73

introduction 69

modules and 78

order of object recreation 82

programs and 78

recreating affected objects 81

recreation failure 82

recreation procedure

for fields or files, changes to 83

for modules or service programs, changes to 85

from within an Application Development

Manager/400 session 85

service programs and 78

impacted modules 78

impacted programs 78

impacted service programs 78

in use status 28

index file

creating based on a physical file 61

creating for SQL 59, 61

installing Application Dictionary Services/400 9

Integrated Language Environment (ILE)

benefits 1

calling other objects

description 90

dynamic calls 90

static calls 90

compilers, ILE 100

data items

sharing between objects 40

definition 1

modules

described 3

procedures, contrasted with 3

object types for 1

OPM compared 1

procedures, described 2

programs

creating 39

programs, described 2

service programs

creating 39

described 3

J

job QDNDCTSYN status description 120

journals

creating 61

displaying 120

K

key fields

defining for physical files 53

description 53

keywords

changing 54

file-level definitions 49

record-level definitions 50

L

last changed source date, finding objects 116

LCKW (Lock Wait) status 120

libraries

adding to a dictionary 32

authority for 28

creating 28

documenting 32

finding documented 36

removing documentation of 34

logical files

creating 56

finding related files 108

M

messages, displaying 29, 31

meta information

creating objects and 6

deleting objects and 6

description 6

migrating

Application Dictionary Services/400 9

dictionaries 9

modules

by last changed source date, finding 116

creating

in Application Dictionary Services/400 40

system's role 39

field changes affecting 78

finding

data areas used by 114

files used by 109

ILE calling structure 97

introduction 90

OPM calling structure 91

that reference files 102, 104

ILE 3

- multiple projects, supporting 37
- multiple sessions
 - example 25
 - starting additional sessions 24
 - switching 25
 - switching tasks 24

N

- naming rules for dictionaries 29
- not complete status 28
- not in use status 28

O

- object types, ILE 1
- objects
 - building
 - introduction 64
 - method 64
 - methods 64
 - by last changed source date, finding 116
 - compile order 77
 - compile order, example 77
 - creating 39
 - creating other types
 - classes, creating 61
 - commands, creating 61
 - journals, creating 61
 - menus, creating 61
 - panel groups, creating 61
 - QUERY Manager forms, queues, creating 61
 - queues, creating 61
 - search indexes, creating 61
 - tables, creating 61

P

- parameters
 - ADM/400 session 125, 127
 - application dictionary 125
 - changing CRTPF 56
 - ENTMOD 40
 - EXPORT 40
 - group 125
 - project 125
 - search path 125
 - starting a debug session 16
 - STRADS command
 - WRKAPPDCT command
- physical files
 - creating, using CRTPF command 56
 - CRTPF parameters, changing 56
 - DDS source, saving 55
 - field definitions 50
 - file-level keyword definition 49

- physical files (*continued*)
 - key field definitions 53
 - record-level keyword definition 50
 - related files, finding 104
- procedures
 - ILE 2, 3
 - non-ILE same name 93
- processing status 28
- programs
 - by last changed source date, finding 116
 - creating
 - ILE 39
 - in Application Dictionary Services/400 40
 - system's role 39
 - finding
 - data areas used by 114
 - files used by 109
 - ILE calling structure 97
 - introduction 90
 - OPM calling structure 91
 - that reference data areas 100
 - that reference files 102, 104
 - non-ILE same name 93
- programs, ILE 2
- projects, supporting multiple 37
- publications, related ix, 131

Q

- QAUDLVL (security auditing level) system value 119, 121
- QDMT library, backing up 122
- QDMT subsystem 25
- QDNDCTSYN job 122
- QDNDCTSYN status
 - LCKW (Lock Wait) 120
 - RUN (Running) 120
 - TIMW (Time Wait) 120
- QDNNOTIFY subsystem 25

R

- rebuilding a dictionary 121
- record formats used in files, finding 111
- record-level keywords, defining 50
- recreating affected objects 81
- recreating objects
 - default options used for 81
 - failure in process 82
 - introduction 81
 - order of recreation 82
 - recreation procedure
 - for fields or files, changes to 83
 - for modules or service programs, changes to 85
 - from within an Application Development Manager/400 session 85

- refreshing
 - automatic 119
 - screen 16
- relative reference levels of fields, example 75
- releasing a held job 122
- reorganizing a dictionary 35
- Report Program Generator (RPG)
 - compiler 100
 - scan RPG source, batch job 90
 - scanning for field names 89
- restrictions, System/38 and System/36 129
- RPG
 - See Report Program Generator (RPG)
- RUN (Running) status 120

S

- same name, programs and non-ILE procedures 93
- scan RPG source
 - batch job 90
 - for field names 89
 - procedure 89
- screen, refreshing 16
- search path 13
- security auditing level (QAUDLVL) 121
- service programs
 - by last changed source date, finding 116
 - creating
 - ILE 39
 - in Application Dictionary Services/400 40
 - system's role 39
 - described 3
 - field changes affecting 78
 - finding
 - data areas used by 114
 - files used by 109
 - ILE calling structure 97
 - introduction 90
 - OPM calling structure 91
 - that reference files 102, 104
 - public interface to 3
- sessions
 - multiple 24
 - switching 25
- SEU
 - See source entry utility (SEU)
- source date last changed, finding objects 116
- source entry utility (SEU)
 - editing DDS source members 80
- SQL
 - See Structured Query Language (SQL)
- Start AppDict Services/400 (STRADS) command 10, 125
- starting a debug session
 - parameters 16
 - Start Debug (STRDBG) command 16

- starting a debug session (*continued*)
 - STRDBG (Start Debug) command 16
- starting a subsystem 25
- starting Application Dictionary Services/400 10
 - Application Development Manager/400 session 12
 - selecting a dictionary 15
 - stand-alone 10
 - using WRKAPPDCT command 14
- static information 5
- status of dictionaries 28
- STRADS (Start AppDict Services/400) command 10, 125
- STRADS command in Application Development Manager/400 session, example 126
- STRDBG (Start Debug) command 16
- strings, searching for 88
- Structured Query Language (SQL)
 - creating a table file 57
 - creating an index file 59
 - creating an index file based on a physical file 61
 - creating index or view files from physical 61
- submit to batch
 - creating objects in dictionary 84
 - recreating objects 84
 - removing documentation from a dictionary 34
- subsystems
 - QDMT 25
 - QDNNOTIFY 25
 - starting 25
 - stopping 25
- supporting multiple projects 37
- switching tasks
 - description 24
 - example 25
- system value
 - QAUDLVL (security auditing level) 119, 121
 - System/38 and System/36 restrictions 129
 - System/38 commands, entering 19

T

- tasks, switching 24
- TIMW (Time Wait) status 120

U

- updating dictionary information 1
- user environments 4, 19
- user-defined options
 - changing 20
 - creating 20
 - example 24
 - showing 22
 - substitution variables 20

W

Work with Active Jobs (WRKACTJOB) command 120

Work with Application Dictionaries (WRKAPPDCT)
command 127

Work with Dictionaries (WRKAPPDCT) command 14

Work with Spooled Files (WRKSPLF) command 90

Work with Submitted Jobs (WRKSBMJOB)
command 122

WRKACTJOB (Work with Active Jobs) command 120

WRKAPPDCT (Work with Application Dictionaries)
command 127

WRKAPPDCT (Work with Dictionaries) command 14

WRKAPPDCT command, example 127

WRKSBMJOB (Work with Submitted Jobs)
command 122

WRKSPLF (Work with Spooled Files) command 90