

MICRO FOCUS

APS FOR z/OS

GETTING STARTED



Copyright © 2002 Micro Focus International Limited.
All rights reserved.

Micro Focus International Limited has made every effort to ensure that this book is correct and accurate, but reserves the right to make changes without notice at its sole discretion at any time. The software described in this document is supplied under a license and may be used or copied only in accordance with the terms of such license, and in particular any warranty of fitness of Micro Focus software products for any particular purpose is expressly excluded and in no event will Micro Focus be liable for any consequential loss.

Animator®, COBOL Workbench®, EnterpriseLink®, Mainframe Express®, Micro Focus®, Net Express®, REQL® and Revolve® are registered trademarks, and AAI™, Analyzer™, Application to Application Interface™, AddPack™, AppTrack™, AssetMiner™, CCI™, DataConnect™, Dialog System™, EuroSmart™, FixPack™, LEVEL II COBOL™, License Management Facility™, License Server™, Mainframe Access™, Mainframe Manager™, Micro Focus COBOL™, Object COBOL™, OpenESQL™, Personal COBOL™, Professional COBOL™, Server Express™, SmartFind™, SmartFind Plus™, SmartFix™, SourceConnect™, Toolbox™, WebSync™, and Xilerator™ are trademarks of Micro Focus International Limited. All other trademarks are the property of their respective owners.

No part of this publication, with the exception of the software product user documentation contained on a CD-ROM, may be copied, photocopied, reproduced, transmitted, transcribed, or reduced to any electronic medium or machine-readable form without prior written consent of Micro Focus International Limited.

Licenses may duplicate the software product user documentation contained on a CD-ROM, but only to the extent necessary to support the users authorized access to the software under the license agreement. Any reproduction of the documentation, regardless of whether the documentation is reproduced in whole or in part, must be accompanied by this copyright statement in its entirety, without modification.

U.S. GOVERNMENT RESTRICTED RIGHTS. It is acknowledged that the Software and the Documentation were developed at private expense, that no part is in the public domain, and that the Software and Documentation are Commercial Computer Software provided with RESTRICTED RIGHTS under Federal Acquisition Regulations and agency supplements to them. Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of The Rights in Technical Data and Computer Software clause at DFAR 252.227-7013 et. seq. or subparagraphs (c)(1) and (2) of the Commercial Computer Software Restricted Rights at FAR 52.227-19, as applicable. Contractor is Micro Focus, 9420 Key West Avenue, Rockville, Maryland 20850. Rights are reserved under copyright laws of the United States with respect to unpublished portions of the Software.

Table of Contents

1	Introduction	5
	Introduction to APS	5
	A Scenario for Using APS.....	5
	Sample Application	7
	Development Steps	10
2	Starting APS	13
3	Beginning Application Development	15
	Define the Development Environment.....	15
	Select APS Application Painters.....	17
4	Defining the Application	19
5	Painting Screens	21
	Learn about the Screen Painter.....	21
	Paint the Screen	24
	Assign Field Attributes.....	26
6	Prototyping the Screen Flow	29
	Create a Screen Flow Prototype	29
	Run the Scenario	31
7	Applying Screen Field Edits	37
	Learn about Field Edits	37
	Assign Field Edits	40

8	Importing Database Definitions	49
	Learn about the Importer Facility	49
	Verify the Database Import	50
9	Defining Program Specifications	55
	Learn About Online Express	55
	Define Program Functions	58
	Map Screen Fields	61
10	Defining Database Access	65
	Learn Database Access through Online Express	65
	Define the Accesses	67
	Customize Database Access	71
11	Generating the Application	75
	Learn about the Generation Process	75
	View Generated Output	77
12	Running the Application	79

1 Introduction

This chapter introduces you to APS for z/OS.

Introduction to APS

Generate MIS applications automatically

APS is a full-function application generator that automates the development and redevelopment of the MIS applications that support your business. With APS you can build simple or complex applications for a variety of IBM production environments. You can generate online and batch applications without manual coding.

Improve development productivity

Productivity improves because you can generate a complete working application without first becoming an expert in IMS, CICS, ISPF, SQL, or any other environment that APS supports. Because APS lets you focus on high-level requirements and specifications, novice developers can quickly generate simple designs with minimal training, and soon build incrementally to complex applications.

APS also encourages developers to share and reuse design modules. It lets you store all application design information in one central location on a network or mainframe, so that multiple users can access that information concurrently. As a result, your application designs always reflect the entire team's current work, and it is easy to share data.

A Scenario for Using APS

Start development with tasks you prefer

APS lets you work in whatever sequence you choose. For example, you can define the user interface before you think about the program logic. Or you can first define the global data that all programs must use. Whether you decide to work top down, bottom up, or middle out, APS lets you proceed from the step you just completed to the one you want to do next. And, if you are maintaining or redeveloping existing applications, APS lets you focus on only those components that must change.

Build applications from high-level specifications It's often best to begin by defining your application's runtime or target environment. When you select an environment, APS handles the necessary implementation details when it generates the application. APS supports the following targets:

Database	IDMS DB, IMS DB, SQL, and VSAM
Data Communications	CICS, IMS DC, and ISPF Dialog

Generate applications for multiple targets If your application must run in several target environments, you can easily specify it to do so. For example, you can generate an application so that it can access more than one kind of database or run as a cooperative processing application across PC and mainframe platforms. In each case, APS generates code that runs in the environments you specify in your high-level application design.

Whenever you are ready, you import into APS the subschemas or tables for your existing databases. Your APS applications can then access these existing databases - first for prototype testing within APS, and then for running the final application.

Prototype the application's look and feel You can next prototype the look and feel of an application, so that end users can review it early in the design process. You then paint the screens that support the user interface you select.

Paint character-based applications APS lets you paint menu and data entry screens that include data entry and text fields. You can also include message fields that let the application communicate errors.

At this stage of prototyping, you and your end users run the screens to ensure that the application user interface meets end user expectations. You can enter data into data entry fields to ensure that they capture all of the required information. You can test various display sequences of character screens to ensure that the screens support an intuitive work flow. Because it is so easy to create, run, and change these prototypes, you can work closely with end users to refine this aspect of the application before you move on to design the underlying details of the application.

Specify application logic APS also simplifies the task of defining an application's processing logic. You can generate online applications from high-level designs using the default logic that APS produces, or you can tailor the logic to your particular requirements. For example, you can tailor the way that APS processes database calls, error routines, or other program functions.

You create batch applications using whatever combination of specification language, user-defined macros, or COBOL II syntax that you prefer. You can combine online and batch programs as you like within a single application.

Automatically generate a working application

When you generate an application, it is ready to install and set up to run in your production environment. Generation produces consistent, high-quality code without run-time modules.

Test run the application

Once you generate an application, you can test run your work within the APS or the MVS environment. Doing so allows you to find features that do not meet expectations, and then modify and retest those features without first setting up the complete target environment that the application will ultimately run in.

To test run character-based applications within APS, you can use the ISPF prototype option to test out your application's program navigation and flow - for example, sending screens, passing control from one program to another, and terminating programs. If you have imported your database definitions, you can test the prototype using test data in your actual SQL or VSAM (but not IMS) database; otherwise you can test the processing logic using data that you enter into screens but do not store.

Sample Application

Learn the basic features of APS

The sample application used in this tutorial introduces you to the primary features of APS. We have selected ISPF/VSAM as the development target for this tutorial because of its popularity. In concept, however, the material covered in this book works for all APS target environments.

Develop a full-function application

The initial application consists of two programs and their associated screens: the Main Menu and the Add Menu. You will add the third program and screen, the Update Menu.

Customer Record Maintenance program

The second program, which we have also supplied for you, lets the end user add parts information and records. The program specifications are:

Program name: APS Add (APSADD)
 Purpose: Add parts information; exit to menu
 Program functions: Add database records, and then exit to menu via F3 key
 Database access: Store
 Record stored: PART-MASTER-REC
 Fields displayed: All fields of a single record
 Associated screen: APS Parts Add (APSA)

Figure 1-2. Customer Record Screen

```

      APS DEMONSTRATION SYSTEM
      PARTS ADD SCREEN

      FUNCTION: X (A-ADD)

      PART NUMBER: XXXXXXXX  DESCRIPTION: XXXXXXXXXXXXXXXXXXXX

      UNITS: XX              BASE PRICE: XXXXXXXX

      DIMENSIONS: XXXXXXXX

XXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      HIT PF3/PF15 TO RETURN TO MENU SCREEN
  
```

Part Inventory List program

The final program lets the end user query, update, and delete part records. Here are the program specifications:

Program name: APS Update (APSUPD)
 Purpose: Query, update, delete multiple part records, and then exit to menu

Program functions: Query database, and then exit to menu via the F3 key
 Database accesses: Obtain, Modify, Delete
 Record accessed: PART-MASTER-REC
 Fields displayed: All fields of a single record
 Associated screen: APS Parts Update (APSU)

Figure 1-3. Parts Update Screen

```

APS DEMONSTRATION SYSTEM
PARTS UPDATE SCREEN

FUNCTION: X (Q=QUERY, U=UPDATE, D=DELETE)

PART NUMBER: XXXXXXXX DESCRIPTION: XXXXXXXXXXXXXXXXXXXXXXXX

UNITS: XX          BASE PRICE: XXXXXXXX

DIMENSIONS: XXXXXXXX

XXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
QUERY REQUIRED BEFORE UPDATE OR DELETE
HIT PF3/PF15 TO RETURN TO MENU SCREEN
    
```

Development Steps

To develop the Parts Update program for this application, you perform the following series of development steps:

- Step 1 Access APS.
- Step 2 Begin application development:
 - Set your project environment.
 - Access the APS Painters.
- Step 3 Create the application view by defining the application and associated programs and other entities with the Application Painter.

- Step 4 Paint the associated screens with the Screen Painter:
- Create the screen mock-ups.
 - Assign field attributes to the screen fields.
- Step 5 Create and run screen flow prototype in the Scenario Painter.
- Step 6 Assign field edits in the Screen Painter to the screen fields.
- Step 7 Import your database definitions via the APS/VSAM File Importer and print a verifying Program DB/DC Report from the APS Documentation Facility.
- Step 8 Define program specifications and navigational logic in Online Express.
- Step 9 Define database access and error handling in Online Express.
- Step 10 Generate the application for the ISPF/VSAM environment from the Application Painter and verify the generation process of the APS Generators.

You need not perform the application development steps in this order. The sequence presented here is simply a guideline that illustrates one of the ways to develop an application. When you become familiar with APS, you can develop your own application development procedure.

2 Starting APS

Goal In this section, you access APS for z/OS.

- Procedure**
- 1 Check with your APS Administrator to ensure that APS is installed at your site.
 - 2 Check with your APS Administrator to ensure that the VSAM files for this application exist.
 - 3 Receive instructions from your APS Administrator on how to access APS at your site. When you complete the access, the APS Main Menu displays.

Checkpoint You have completed the signon procedure for APS. You are now ready to define the application development environment.

Help The file APS software PDS CNTL contains JCL to create the VSAM files. To create the VSAM files yourself, do the following:

- Edit the APS software PDS CNTL(APSDEMO) member.
- Follow the instructions in the JCL.
- Submit the job.

3 Beginning Application Development

In this chapter, you begin developing an application by defining the development environment and selecting an APS Application Painter.

Define the Development Environment

Goal In this section, you specify where your application will be physically located.

You specify the application path, project, group, and DDIFILE (the staging area for symbols) to point to the correct files. Use the APS Options Menu to define the development environment and to change the default values assigned at installation time.

Procedure Start this procedure where the last one ended - in the APS Main Menu.

Figure 3-1. APS Main Menu

```

----- APS Main Menu -----
OPTION ==>

0 - Options
1 - Painters
2 - Dictionary Services
3 - Prototype Execution
4 - Utilities
5 - Version Control System

C - Changes in APS for z/OS
T - APS Tutorial
  
```

- 1 If you want to view the definitions that APS assigns to your function keys, type **keys** in the **Option** field and press Enter. Press F3 to return to the Main Menu.

- 2 To transfer to the Options Menu from the Main Menu, type **0** in the **Option** field and press Enter. The Options Menu screen displays.

Figure 3-2. APS Options Menu

```

----- APS Options Menu -----
OPTION ==> █

    0 - Reset Options
    1 - Generator Options
    2 - Project Group Environment
    3 - Precompiler Options
    4 - Report Options
    5 - DB2 Bind Options
    6 - Job Card Options
    7 - IDMS Options
    8 - International Options
  
```

- 3 To transfer to the screen where you can define your environment, type **2** in the **Option** field and press Enter. The Project Group Environment screen displays.
- 4 In the Project, Group, and DDIFILE fields, enter the project, group, and DDIFILE specified when you installed APS.
- 5 Press **Enter** to process changes.
- 6 Press **F3** to return to APS Options Menu.
- 7 Press **F3** to return to APS Main Menu.

Checkpoint You are now ready to access the Application Painters where you will do the actual application development.

Help If your cursor is anywhere in the active screen, press the Home key to quickly position it on the **Command** field.

Hints The status line at the bottom of APS screens shows you the APS status (Input or Working), your cursor position (row and column numbers), and whether the insert mode is on or off.

You can use the Options menu to change options set at installation as follows:

- Define options for the generation process.
- Reset profile variables to original installation values.

- Vary selected precompile parameters.
- Define characteristics for documentation reports.
- Specify BIND options for DB2.
- Establish job cards.

Select APS Application Painters

Goal In this section, you enter the Application Painters, and you access the APSAPPL application.

The Application Painters help you quickly define screens, data structures, and programs, and then relate these items to databases. You perform all of your development for this application in the Application Painters.

Procedure Start this procedure where the last one ended - in the APS Main Menu. You transfer to the APS Painter Menu, from which you can access any:

- APS entity
 - ISPF data set
- 1 To transfer to the Painter Menu, type **1** in the **Option** field and press Enter. The APS Painter Menu displays.
 - 2 Type **e** in the **Command** field to create or edit the application.
 - 3 If AP (the default Type) does not appear in the **Type** field, type **ap** in the **Type** field to indicate Application Painter.
 - 4 To select the tutorial application, type the application name, **apsappl**, in the **Member** field.
 - 5 Press Enter. The APS Application Painter screen displays.

Figure 3-3. APS Painter Menu

```

COMMAND ==> e

E - Edit entity                C - Create Like
B - Browse entity            D - Delete entity
P - Print APS report         R - Rename entity

      TYPE ==> AP
      MEMBER ==> apsappl_ (Blank for member selection list, if no option
      NEWNAME ==>          (If option "C" or "R" selected)
      SCREEN SIZE ==> MOD2

TYPE SELECTION:
AP - Application              SC - Screen
PG - Program                 RP - Report
DS - Data Structure          CN - Scenario

```

Checkpoint You are now ready to begin the actual application development. In this tutorial, you add an additional screen and program to the APSAPPL application.

Help You can enter your commands and information in either upper or lower case. We usually show the information you enter in lower case to distinguish this text from the screen text and previously entered information.

Hints Selecting edit mode on the APS Painter Menu allows you to create or modify any application component. The default mode is browse mode, which is for viewing only.

From the APS Painter Menu you can:

- Create, modify, browse, delete, or rename an entity.
- Display an alphabetical listing of entities.
- Print reports.

4 Defining the Application

Goal In this section, you view the application system components and their relationships in the Application Painter. You then add the third application program and screen and the program subschema.

The Application Painter lets you quickly:

- Name the components of the working application: programs and their associated screens and data structures, reports that document the application, subschemas or PSBs that define the database records to the application definition, and macro library members that contain macro code for program use.
- Define the component relationships: for example, which screens are associated with which programs, which data structures are global to the application and which are local to a specific program.
- Specify the database and data communication environments where the application will run.

Procedure Start this procedure where the last one ended - in the APS Application Painter, where the APSAPPL application is defined.

Figure 4-1. APSAPPL Application Definition

COMMAND ===>				SCROLL ===> CSA			
DC ===>	ISPF			AUTHOR ===>	USERNAME		
DB ===>	USAM			SCREEN SIZE ===>	MOD2		
-LINE-	PROGRAMS	SCREENS	IO	REPORTS	DATA STR TY	SBSC/PSB	USERMACS L
000001	APSMNU	APSM	IO				
000002	APSADD	APSA	IO			APSADD	
	apsupd	s apsu	io			apsupd_	
END							

- 1 To insert a blank line in the application definition, type **i** (ISPF Insert command) over the **000002** line number as follows: **i00002**. Press Enter. This command creates a blank line and positions the cursor on it.

- 2 Type the new program name, **apsupd**, the associated screen, **apsu**, and the associated subschema, **apsupd**, as shown in *Figure 4-1*.
- 3 Type **io** to indicate that the screen is for both input and output.
- 4 Press Enter twice.
- 5 Type **s** in the selection field to the left of the APSU screen.
- 6 Press Enter to transfer to the Screen Painter.

Checkpoint You have completed the application definition. You are now ready to set your screen design options and paint the Parts Update screen in the APS Screen Painter.

Help The Application Painter has an online help facility that you can select from the action bar, or access from the screen by pressing PF1.

Hints After you define your application components and relationships, you typically use the Application Painter to transfer to other APS Painters, print reports, and generate, compile, and link your application entities.

5 Painting Screens

In this chapter, you learn about the Screen Painter, paint a screen and assign field attributes.

Learn about the Screen Painter

Goal In this section you learn what the APS Screen Painter does, and you set screen design options.

The APS Screen Painter lets you paint character-based screens that are intuitive and easy to use. You first paint text, input/output fields, and then blocks of fields that accept multiple record occurrences. You then specify field names, field attribute and edit criteria, and generation parameters. The APS Generators retrieve this screen information from the Application Dictionary to produce native screen source code.

A character screen consists of fields and blocks of fields that you paint on a blank screen. You paint two types of fields in the APS Screen Painter:

- Input/Output (I/O) fields that let end users view, add, update, and delete information. You paint I/O fields by typing a string of Xs.
- Text fields that display text, such as prompts for I/O fields, column headings, screen headings, section headings, and explanatory text. You paint text fields by typing any text you want.

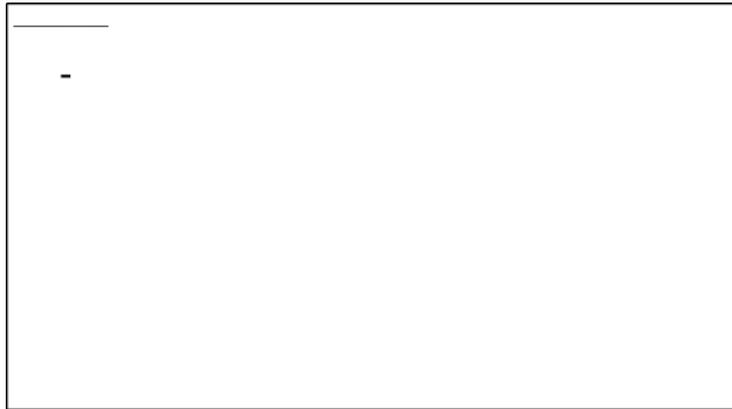
Using a series of related screens in the Screen Painter, you can also:

- Assign field attributes, such as field protection, brightness, cursor positioning, and color.
- Assign field edits, such as an internal picture, edit mask, or date format, or test for a range of valid values the end user inputs into I/O fields.

The Screen Painter also provides editing and design options to help you paint the screen. For example, you can specify where the **Command** field automatically appears on your screen, and can determine whether your text displays in upper case, lower case, or both.

Procedure Start this procedure where the last one ended - in the APS Screen Painter. When the Screen Painter first appears, it is blank except for a line indicating the **Command** field.

Figure 5-1. Creating a Screen



-
- 1 Type profile on the line indicating the **Command** field. The Profile screen displays showing the current values in your user profile of screen design options.

Figure 5-2. Access Your User Profile



2 Define the following characteristics for your session:

Command location	Type top , so the Command field always appears in the upper-left corner, or type bottom for the lower-left corner, if you prefer.
Caps on/off	Type off , so text fields remain in the case (upper or lower) you enter them.
Nulls on/off	Type on , so you can insert data directly into a row.
Keys on/off	Type on , so the PF key definitions display at the bottom of your screen.

3 Press PF3 to return to your blank screen.

Checkpoint You have set your screen design options and are ready to paint the Parts Update screen.

Help The screen design options you select remain in effect for all application screens until you change them, either in the current editing session or a subsequent one.

The Screen Painter has an extensive help facility that you can select from the action bar, or access from your screen by pressing PF1.

Hints Other functions you can perform in the Screen Painter are:

- Paint blocks of fields that accept multiple record occurrences of one or more records.
- Select I/O fields, complete with definitions, attributes, text prompts, and edits, from the APS Data Element Facility.
- Tailor screen generation for a particular target environment.

Paint the Screen

Goal In this section, you paint the text and I/O fields for the Parts Update Screen, paint a field for system messages, paint a field to save the database key, and assign a title to the screen.

When you paint a screen, you generally:

- Design the layout of the screen, with text and input/output (I/O) fields that appear as Xs, as shown in *Figure 5-3*. This layout is called a screen mock-up.
- Paint an I/O field for system messages. APS requires a system message field for any screen that assigns field edits in order to receive screen field edit messages. The system message field is the last I/O field on the screen in *Figure 5-3*.
- Paint a savekey field for records that are read in one execution of the program and reread in the next execution. For example, in a program that updates the database, such as this one:
 - The first execution of the program reads the record and displays it on the end user's screen.
 - The second execution of the program may reread the record before updating it. To do this, APS must save the key of the record when it reads it for the first time.

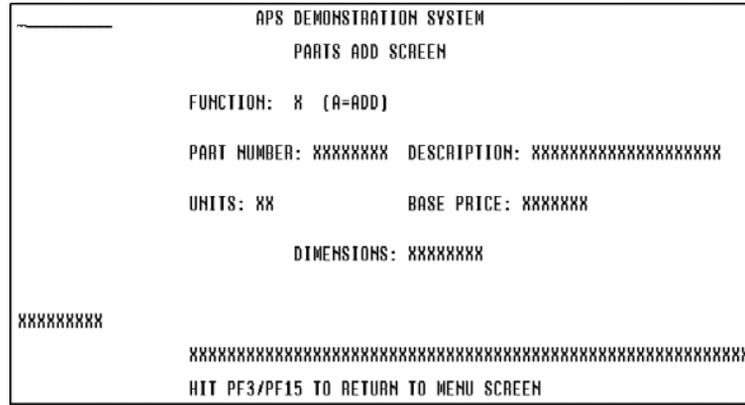
The savekey field is the unlabeled I/O field of nine bytes that precedes the system message field in *Figure 5-3*.

- Assign a screen title that appears in all generated APS documentation and in the screen flow scenario prototype, which you perform in the next chapter, Prototype the Screen Flow.

Procedure Start this procedure where the last one ended - on a blank screen in the Screen Painter. *Figure 5-3* illustrates the Parts Update screen to paint.

- 1 Type the APSU screen layout exactly as shown in *Figure 5-3*.
- 2 Press Enter.

Figure 5-3. Paint a Screen Mock-Up



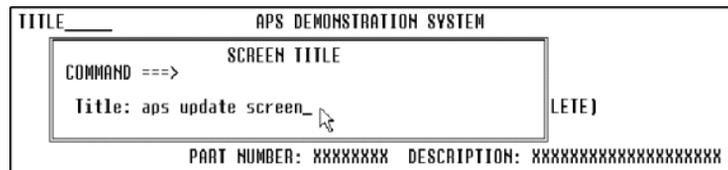
3 To assign a screen title, type **title** in the **Command** field.

Figure 5-4. Enter the TITLE Command



4 Press Enter to display the Screen Title screen.

Figure 5-5. Assign a Screen Title



- 5 Type the screen title, as shown in *Figure 5-5*.
- 6 Press PF3 to save your work and return to the Parts Update screen-painting screen.

Checkpoint You have painted your screen mock-up and are ready to assign attributes to the I/O fields.

Help To change a field length, move the cursor to the Xs designating the field and type in your changes. You can space over or delete the Xs representing the field, or extend the field with more Xs.

To delete screen text, use the Delete key to erase all of part of a field or row.

Hints The screen title does not appear on your screen mock-up; it displays in the Scenario Painter when you test the screen.

Assign Field Attributes

Goal In this section, you assign various attributes to each I/O field.

The APS Screen Painter supports full and extended 3270 attribute capabilities, including:

- Color
- Underline, blinking, and reverse video features
- Cursor positioning when the screen displays to the end user
- Bright and dark intensity
- Numeric keyboard locking
- Field protection
- Assignment of initial value
- Light pen sensitivity

For example, you can enhance the presentation of I/O and text fields by changing the default values for attributes such as intensity, color, and underlining. For I/O fields, you can modify attributes to protect a field from data entry and position the cursor on a particular field.

Additionally, you can change the APS default field name to something descriptive to the application.

APS assigns default attribute values to each field for you. You can quickly override the default by typing over the value.

- Procedure** Start this procedure where the last one ended - on the Parts Update screen painting screen.
- 1 Type **fa** in the **Command** field and press Enter. The Field Attribute screen displays.
 - 2 For the **function** field, type **function** in the **Name** field to change the default field name.
 - 3 For the **Part Number** field, type **part-nbr** in the **Name** field.
 - 4 For the **Description** field, type **short-desc** in the **Name** field.
 - 5 For the **Units** field, type **units-instock** in the **Name** field.
 - 6 For the **Base Price** field, type **base-price** in the **Name** field.
 - 7 For the **Dimensions** field, type **dimensions** in the **Name** field.
 - 8 For the next field, the **savekey** field, do the following:
 - a Type **savekey** in the **Name** field.
 - b Type **p** in the **Ty(pe)** field to change the field from unprotected to protected, so the end user cannot enter data.
 - c Type **d** in the **In(tensity)** field to change the field from normal intensity, to dark (hidden), so the field does not display on the screen that the end user sees.
 - 9 For the **system message** field, do the following:
 - a Type **sysmsg** in the **Name** field.
 - b Type **p** in the **Type** field to protect the field from data entry.
 - c Type **b** in the **Intensity** field to change the field from normal intensity to bright (highlighted).
 - 10 Press PF3 to save the system message field attributes. Press PF3 again to save the screen design and return to the Application Painter.

- Checkpoint** You have completed screen painting. You are now ready to prototype your screen flow where, in an actual application development cycle, you demonstrate the application to your end user.

Hints An I/O field name can have a maximum of 16 characters. Text fields do not have names because programs do not reference them.

If you give a screen field the same name as its corresponding database field, APS Online Express automatically maps the relationship for you, prefixing the field name with the screen name; otherwise you must map the screen field to the database in your program.

If the same field appears on several screens, give it the same name on each screen. APS lets you pass data between identically named fields on different screens during scenario prototyping and ISPF prototyping.

6 Prototyping the Screen Flow

In this chapter, you create a screen flow prototype and run the scenario.

Create a Screen Flow Prototype

Goal In this section you create a screen flow prototype in the APS Scenario Painter.

Before you generate your screens, you can review their design and flow with the end user in the APS Scenario Painter. You can define a sequence of screens, called a scenario, enter data in those screens, and display the screens to the end user.

By using the Scenario Painter early in the development process, you can work with end users to precisely define screen format and sequence before the development efforts begin.

The Scenario Painter automatically creates a scenario by displaying a list of your application screens, where you can reorder the screens as you like, delete screens, or add new screens (whether or not they have been painted). You can also enter descriptive information.

Procedure Start this procedure where the last one ended - in the Application Painter.

- 1 To access the Scenario Painter, type **scen apsappl** in the **Command** field of the Application Painter screen and press Enter. The Scenario Painter screen for the APSAPPL application displays.

Figure 6-1. Access the Scenario Painter

```

EDIT --- SCENARIO: APSAPPL ----- COLUMNS 001 0
COMMAND ==> retitle_ SCROLL ==> CSA
- LINE - - SCREEN - - - - - SCREEN TITLE ----- -- USER COMMENT --
***** TOP OF DATA *****
000001 APSM
000002 APSA
000003 APSU
***** BOTTOM OF DATA *****

```

- To display the titles for the screens in the application, including the one you previously created in the Screen Painter with the Title command, type **retitle** in the **Command** field and press ENTER. The Screen Painter information displays.

Figure 6-2. Display Screen Titles

```

EDIT --- SCENARIO: APSAPPL ----- COLUMNS 001 0
COMMAND ==> run_ SCROLL ==> CSA
- LINE - - SCREEN - - - - - SCREEN TITLE ----- -- USER COMMENT --
***** TOP OF DATA *****
000100 APSM APS MAIN MENU SCREEN
000200 APSA APS PARTS ADD SCREEN
000300 APSU APS UPDATE SCREEN
***** BOTTOM OF DATA *****

```

- To save your prototype definition, enter **save** in the **Command** field.

Checkpoint You have created the screen flow prototype and are ready to run it.

Help For a new prototype definition, the Scenario Painter lists the screens as they appear in the application definition.

To reset the screen flow to its sequence at the beginning of the session, type **reset** in the **Command** field.

The Scenario Painter has an extensive help facility that you can select from the action bar, or access from your screen by pressing PF1.

Hints You can enter descriptive information about each application screen by typing the desired text in the **User Comment** field.

To modify the scenario in this tutorial, use the ISPF I(nsert), D(elete), C(opy), and M(ove) line commands to reorder, insert, and delete screen names until the prototype represents the scenario you want to test. A prototype definition can include up to 160 screens.

To describe the screen for the end user, enter text in the **User Comments** field. For example, a user comment might identify the varying conditions under which the same screen displays multiple times.

Run the Scenario

Goal In this section, you run the scenario sequentially, display descriptive information about the scenario, and simulate the movement of data.

The Scenario Painter offers data simulation capabilities that allow you to quickly establish and demonstrate online application flow. During prototype sessions, you can capture, save, and reuse data to simulate realistic data movement to the end user. All of the data that you enter automatically becomes available to other screens and displays in identically named fields.

While running a scenario, you can display descriptive information on the last line of the screen, such as the screen name, scenario sequence number, total number of scenario screens, screen title, and any descriptive text entered in the **User Comment** field. This information is particularly useful when running large scenarios.

Procedure Start this procedure where the last one ended - in the Scenario Painter.

- 1 To begin the scenario, type **run** in the **Command** field and press Enter. The first screen in the scenario, APSM, displays. Note that Xs indicate the I/O fields.

Figure 6-5. Eliminate the Xs from I/O Fields

```

APS DEMONSTRATION SYSTEM
MENU SCREEN

ENTER OPTION ==> 1_
                    |
                    v
1 --- ADD
2 --- QUERY, UPDATE, DELETE

HIT PF3/PF15 TO EXIT APPLICATION

APSM # 1 OF 3 APS MAIN MENU SCREEN

```

-
- 4 To simulate what the end user must do to select the ADD screen, type **1** in the **Option** field of the Menu Screen and press Enter. The Parts Add Screen displays.

Figure 6-6. Enter Sample Data

```

-
APS DEMONSTRATION SYSTEM
PARTS ADD SCREEN

FUNCTION: a (A=ADD)

PART NUMBER: 11111   DESCRIPTION: widgets
UNITS: 5             BASE PRICE: $100.00
DIMENSIONS: 2" x 2"

HIT PF3/PF15 TO RETURN TO MENU SCREEN
APSA # 2 OF 3 APS PARTS ADD SCREEN

```

-
- 5 Enter sample data as shown in *Figure 6-6*, or enter your own data.
 - 6 Press Enter to display the final screen in the scenario, the APS Update Menu. If desired, enter sample data.

Figure 6-7. Display the Last Screen

```

-
      APS DEMONSTRATION SYSTEM
      PARTS ADD SCREEN

      FUNCTION: a (A=ADD)

      PART NUMBER: 11111   DESCRIPTION: widgets

      UNITS: 5           BASE PRICE: $100.00

                        DIMENSIONS: 2" x 2"

      HIT PF3/PF15 TO RETURN TO MENU SCREEN
APSA # 2 OF 3 APS PARTS ADD SCREEN
  
```

- 7 To turn off the data simulation, press the Home key to position the cursor in the upper left corner. Then type **dataoff** and press Enter. The screen displays in mock-up format.

Figure 6-8. Turn Off Data Simulation

```

-
      APS DEMONSTRATION SYSTEM
      PARTS UPDATE SCREEN

      FUNCTION: X (Q=QUERY, U=UPDATE, D=DELETE)

      PART NUMBER: XXXXXXXX DESCRIPTION: XXXXXXXXXXXXXXXXXXXXXXXX

      UNITS: XX           BASE PRICE: XXXXXXXX

                        DIMENSIONS: XXXXXXXX

XXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
QUERY REQUIRED BEFORE UPDATE OR DELETE
HIT PF3/PF15 TO RETURN TO MENU SCREEN
  
```

- 8 To exit the scenario, press Enter.
- 9 Press F3 to return to the Application Painter.

Checkpoint You have demonstrated the screen flow prototype. The next step in this tutorial is to assign field edits in the screen fields.

Help To store the current data entered in this session for use in future prototyping sessions, while running the scenario, type **save** in the upper left corner of the screen.

The READ command displays data saved by the most recently executed SAVE command and also executes the DATA command.

After viewing the last screen in the scenario, press Enter to exit the Scenario Painter.

To modify the viewing sequence of the screens, type any of the following commands, followed by a space, in the top left corner of the displayed screen.

Command	Description
start, first	Display the first screen in the prototype. When you press Enter, the second screen displays, and so on.
last	Display the last screen in the prototype.
end, can, quit	Terminate the prototype and return to the invoking screen.

Hints To display screens consecutively, press Enter repeatedly until all screens display. If a screen named in your prototype is not yet painted in the Screen Painter, a message displays that information.

7 Applying Screen Field Edits

In this chapter, you learn about field edits and assign them to the fields on your screens.

Learn about Field Edits

Goal In this section you learn about what field edits are and the different types of edits you can create. You also access the Field Edit Facility in the Screen Painter.

Field edits let you define the display and storage characteristics for I/O fields. Field edits can validate input data and format that data for storage and output. You can assign characteristics, such as an internal picture, output picture, edit mask, or date format. Or, you can use field edits to test for specific values or a range of values.

The internal data representation specifies storage characteristics for data in a field. Input and output data representations let you specify the type of data that users can enter or that a field can display. For example, an input data representation for a field may permit a user to enter numbers from 1 to 1,000; an output data representation may require that data display a dollar sign, decimal point, and two places following the decimal point.

Depending on the type of field you create - character, numeric, or date - you assign field edits that define the data representations for the field. APS supports the following field edits:

- Internal field edit values specify the COBOL picture characteristics for storing data the end user enters. These include:
 - Alphabetic, character, or numeric data type
 - Internal length
 - Binary or packed format

- Sign specification
- Right justification
- Input field edit values specify format and data requirements that the end user must adhere to when entering data into the field. These include:
 - Required data
 - Input mask
 - Julian, Gregorian, or system date type
 - Minimum/maximum input requirements
 - Testing for blank spaces or special characters
 - Testing for numeric data
 - Testing for specific values
 - Zero-fill when blank
- Output field edit values specify the format requirements for displaying the data. These include:
 - Output mask
 - Output picture
 - Julian, Gregorian, or system date type
 - Right justification
 - Commas
 - Zero suppression
 - Floating, leading, and trailing symbols
- Values or conversion values ensure only certain values are entered. You can test for:
 - A specific value or range of values for input data
 - Conversion values for input and output data representation to define how data is stored and converted for output
- Application edit routines specify additional edits or tests for input or output data. Edit routines can be paragraphs, subprograms, or

APS macros that you create. Or, you can select a predefined application edit routine from a centralized listing of edit routines maintained by your APS Administrator.

When users fail to enter data correctly, you can display an error message that explains the problem. You can define two error messages, one for when the end user enters invalid data, and one for when the end user neglects to enter data that is required.

Procedure Start this procedure where the last one ended - in the Application Painter.

- 1 On the Application Painter screen, select the APSU screen by entering **s** in the **selection** field and press Enter. The Parts Update screen displays.
- 2 To access the Field Edit Facility, type **fe** in the **Command** field and press Enter. The Field Selection screen displays.

Checkpoint You are in the Field Edit Facility and are ready to assign edits to each of your screen fields.

Help The Screen Painter has an extensive help facility for field edits that you can select from the action bar, or access from your screen by pressing F1.

Hints You can also code your own edit routines and apply them to multiple screens across any number of application systems. APS field edits ensure that entries match specified definitions. Some fields, however, require specialized testing. For example, if a field has alternate formats, no single field edit can confirm the validity of all possible entries. In such a case, you can write an application edit that verifies all legal entries. Or, you can select a predefined edit from a centralized application edit listing.

Error messages can be global messages that display for all screen fields, or specific messages that display for individual fields. Messages defined for a specific field override the global messages. You can also define conditions for bypassing input edits under certain conditions.

Assign Field Edits

Goal In this section, you assign the following field edits to the Parts Update screen.

Field	Type	Edit(s)
FUNCTION	Character	Require data entry.
PART-NBR	Numeric	Store data as an 8-digit number. Require data entry. Remove special characters . Define output picture as ZZZZZZZ9 .
UNITS-INSTOCK	Numeric	Store data as a 2-digit number. Allow only data between 0 and 50. Move zeros to field if data not entered. Remove special characters. Specify an error message that says VALID VALUES FOR UNITS ARE 0 THROUGH 50 . Define output picture as Z9 .
BASE-PRICE	Numeric	Store data as a 4-digit number with 2 decimal places. Move zeros to field if data not entered. Remove special characters. Define output picture as ZZZ9.99 .

Procedure Start this procedure where the last one ended - in the Field Selection screen. The Field Selection screen displays the screen fields for your screen, and lists any edits specified.

Figure 7-1. Summary Field Edit Listing

```

COMMAND ==>
                                                    SCROLL==> PAG
I - Specify input edits          O - Specify output edits
U - Specify values or conversions P - Specify an internal picture
S - Select APS edit menu        D - Delete a field's edits

FIELD-NAME      LEN  ROW  COL  INTERNAL  INPUT  OUTPUT  VALUES
FUNCTION        001  006  031
PART-NBR        000  009  033  *         *         *
SHORT-DESC      020  009  056
UNITS-INSTOCK   002  012  027  *         *         *
BASE-PRICE      007  012  055  *         *         *
DIMENSIONS      000  015  042
SALEKEY         009  019  002
SYSMSG         060  020  020
***** BOTTOM OF DATA *****

```

- 1 To select the Function field for edit specification, type **s** in the selection field to the left of FUNCTION and press Enter, as shown in *Figure 7-1*. The Edit Selection screen indicates the edits currently specified for the field.

Figure 7-2. Select Field to Assign Edits

```

COMMAND ==>

SCREEN FIELD NAME : FUNCTION      LEN 001  ROW 006  COL 031
INTERNAL PICTURE  ==>
INPUT EDITING     ==>
  ERROR PROCESSING ==>
  APPLICATION EDITS ==>
OUTPUT EDITING   ==>
  APPLICATION EDITS ==>
VALUES OR CONVERSIONS ==>
DATE EDITING     ==>

```

- 2 To specify input criteria for the field, type **s** in the **Input Editing** field and press Enter. The Character Input screen displays, where you can specify input edits for any field with an internal picture defined as alphabetic or character.

Figure 7-3. Specify Input Criteria for Character Fields

SCREEN FIELD NAME	FUNCTION	LEN	ROW	COL
INTERNAL PICTURE	==>	X(01)		
REQUIRED	==>	s		
INPUT MASK	==>			
MINIMUM INPUT	==>			
MAXIMUM INPUT	==>			
NO EMBEDDED SPACES	==>			
NUMERIC TEST	==>			
VALUES or CONVERSIONS	==>			
APPLICATION EDITS	==>			
ERROR PROCESSING	==>			

- 3 To require data entry for this field, type **s** in the **Required** field and press Enter.
 - 4 Press F3 twice to save the edit and return to the Field Selection screen, where you see the edit for the **Function** field reflected.
 - 5 To select the **Part Number** field for edit specification, type **s** in the selection field to the left of PART-NBR and press Enter. The Edit Selection screen displays.
 - 6 To automatically access the applicable screens for assigning the required edits, do the following:
 - Type **s** in the **Internal Picture** field.
 - Type **s** in the **Input Editing** field.
 - Type **s** in the **Output Editing** field.
- Then press Enter. The Internal Picture screen displays.

Figure 7-4. Generate a COBOL Picture for Internal Storage

```

COMMAND ==>

SCREEN FIELD NAME : PART-NBR          LEN 008  ROW 009  COL 033

DATA TYPE ==> n          9(08)
  A - Alphabetic  C - Character  N - Numeric  G - EGCS

INTERNAL LENGTH ==> 8  (Characters or Digits left of decimal)

NUMERIC OPTIONS:
  DECIMAL PLACES ==>          SIGNED ==>
  COMP {BINARY} ==>          SIGN LEADING ==>
  COMP-3 {PACKED} ==>        SIGN SEPARATE ==>

CHARACTER OPTIONS:
  JUSTIFIED RIGHT ==>

```

- 7 To store the data as an 8-digit number, type **n** in the **Data Type** field and type **8** in the **Internal Length** field. Optionally press **Enter** to view the edits.
 - 8 Press F3 to save these edits. Because you defined the field as numeric and you selected Input Editing on the Edit Selection screen, the Numeric Input screen automatically displays.
-

Figure 7-5. Specify Input Criteria for a Numeric Field

```

COMMAND ==>

SCREEN FIELD NAME : PART-NBR          LEN 008  ROW 009  COL 033

INTERNAL PICTURE ==>          9(08)

INPUT MASK ==>
NUMERIC DE-EDIT ==> s

REQUIRED ==> 5  ZERO WHEN BLANK ==>
MINIMUM DIGITS ==> MINIMUM DECIMALS ==>
MAXIMUM DIGITS ==> MAXIMUM DECIMALS ==>

VALUES or CONVERSIONS ==>

APPLICATION EDITS ==>

ERROR PROCESSING ==>

```

- 9 To remove special characters from the input data, type **s** in the **Numeric De-Edit** field.
- 10 To require data entry for this field, type **s** in the **Required** field. Optionally press **Enter** to view the edits.

- 11 Press F3 to save these edits. Because you defined the field as numeric and you selected Output Editing on the Edit Selection screen, the Numeric Output screen automatically displays.

Figure 7-6. Specify Output Criteria for a Numeric Field

```

COMMAND ==>
SCREEN FIELD NAME : PART-NBR          LEN 008  ROW 009  COL 033
INTERNAL PICTURE ==>          9(08)
OUTPUT MASK      ==>
OUTPUT PICTURE OPTIONS:
  OUTPUT PICTURE ==>          zzzzzzz9
  INSERT COMMA(S) ==>
  ZERO SUPPRESSION ==>
  FLOATING SYMBOL ==>          { $, +, - }
  FIXED LEADING SYMBOL ==>    { $, +, - }
  FIXED TRAILING SYMBOL ==>   { +, -, CR, DB }
CONVERSIONS     ==>
APPLICATION EDITS ==>

```

- 12 To assign the COBOL output picture, type **zzzzzzz9** in the **Output Picture** field. Optionally press Enter to view the edits.
- 13 Press F3 twice to save these edits and return to the Field Selection screen, where you will see your edits reflected.
- 14 To select the **Units In Stock** field for edit specification, type **s** in the selection field to the left of UNITS-INSTOCK and press Enter. The Edit Selection screen displays.
- 15 To automatically access the applicable screens for assigning the required edits, do the following:
 - Type **s** in the **Internal Picture** field.
 - Type **s** in the **Input Editing** field.
 - Type **s** in the **Output Editing** field.
 Then press Enter. The Internal Picture screen displays.
- 16 To store the data as a 2-digit number, type **n** in the **Data Type** field and type **2** in the **Internal Length** field. Optionally press Enter to view the edits.

Figure 7-8. Define an Error Message for Invalid Data

```

COMMAND ==>

SCREEN FIELD NAME : UNITS-INSTOCK      LEN 002  ROW 012  COL 027
  Enter message for field in error below:
> valid values for units are 0 through 50

  Enter message for a required field not entered below:
>

Using TP-ATTR attribute syntax, enter any valid attribute combination below
==> pos+brt_

```

- 24 To assign the error message, type **valid values are 0 through 50** below the **Enter Message for Field in Error** line.
- 25 To position the cursor at the field when invalid data is entered, and to assign a bright intensity to the error message, type **pos+brt** below the **Using TP-ATTR Attribute Syntax** line. Optionally press Enter to view the edits.
- 26 Press F3 twice. The Numeric Output screen automatically displays.
- 27 To assign the COBOL output picture, type **z9** in the **Output Picture** field. Optionally press Enter to view the edits.
- 28 Press F3 twice to save these edits and return to the Field Selection screen, where you will see your edits reflected.
- 29 To assign a field to display the error message you assigned, type **parm** in the **Command** field and press Enter.
- 30 Type **sysmsg** in the **Error Message Display Field** field. Optionally press Enter to view the edit.
- 31 Press F3 to return to the Field Selection screen.
- 32 To select the **Base Price** field for edit specification, type **s** in the selection field to the left of BASE-PRICE and press Enter. The Edit Selection screen displays.
- 33 To automatically access the applicable screens for assigning the required edits, do the following:
 - Type **s** in the **Internal Picture** field.

- Type **s** in the **Input Editing** field.
- Type **s** in the **Output Editing** field.

Then press Enter. The Internal Picture screen displays.

- 34 To store the data as a 4-digit number with 2 decimal places, type **n** in the **Data Type** field, type **4** in the **Internal Length** field, and type **2** in the **Decimal Places** field. Optionally press Enter to view the edits.
- 35 Press F3 to save these edits. The Numeric Input screen automatically displays.
- 36 To remove special characters from the input data, type **s** in the **Numeric De-Edit** field.
- 37 To move zeros to the field if data is not entered, type **s** in the **Zero When Blank** field. Optionally press Enter to view the edits.
- 38 Press F3. The Numeric Output screen automatically displays.
- 39 To assign the COBOL output picture, type **zzz9.99** in the **Output Picture** field. Optionally press Enter to view the edit.
- 40 Press F3 twice to save these edits and return to the Field Selection screen, where you will see your edits reflected.
- 41 Press F3 repeatedly until you return to the APS Main Menu.

Checkpoint You have assigned the field edits to your screen fields. You are ready to import your database definitions, a required step before you define the program.

Help You can delete field edits in one of the following ways:

- To delete all field edits for all fields on the screen, access the Field Selection screen, and then type **delete all** in the **Command** field and press Enter. At the prompt, verify that you want to delete all field edits.
- To delete all field edits for a specific field, access the Edit Selection screen for the field, and type **d** next to the field name. At the prompt, verify the deletion.
- To delete a specific field edit for a specific field, access the Edit Selection screen for the field, and type **d** to the right of the field edit name and press F3. At the prompt, verify the deletion.

Hints To copy edits from another field, access the Edit Selection screen for the field you are copying field edits to, and then enter **copy** in the **Command** field. The Copy Function screen displays. Enter the field name you are copying edits from; it must be the same length as the current field. The current field inherits the edits of the copied field, and loses any prior edits.

On the Parm screen you can define two error messages, one for when the end user enters invalid data, and one for when the end user neglects to enter data that is required that display for all screen fields. Any messages defined for a specific field then override these global messages. You can also define conditions for bypassing input edits under certain conditions on this screen.

8 Importing Database Definitions

In this chapter, you learn about the Importer Facility and import a database.

Learn about the Importer Facility

Goal In this section, you learn about importing database definitions using the APS Importer Facility.

The Importer Facility translates information about your database definitions and their associated copybook records into a format usable for generating and precompiling through APS. The APS Importer Facility transfers database information such as data definitions and/or database subschemas to make them available to your APS programs. You can import IMS PSBs and DBDs, SQL DDL statements, VSAM files, and IDMS subschemas.

If necessary, your APS programs can access multiple database targets. In that case, you can create and import two separate subschemas. For example, you can import both VSAM file information and a subschema for SQL, and then combine the two into one subschema.

When you import your database definitions, the Database Definition Interface (DDI) formats the database information to use with APS programs. The Importers:

- Extract information from your database definition.
- Load extracted information into the DDIFILE.
- Generate a DDISYMB file for use by Online Express and the appropriate APS Generator.

- Generate record description copybooks of SQL DDL statements that contain database and COBOL descriptions of each table or view in the imported subschema.

To import the VSAM file definitions for this tutorial, you first write DDI statements and then import the file definitions. The DDI statements associate the VSAM files with their COBOL copylib record descriptions.

The VSAM Importer translates information about VSAM files and their associated copylib record into a format usable for generating and precompiling through APS. For each VSAM file record, there is a copylib file that contains a COBOL record description. These copylib files are in your *project.group.COPYLIB*.

We imported the VSAM database for you. You can verify the import process.

Verify the Database Import

Goal In this section, you verify the import process for this tutorial by printing the Program DB/DC Report, one of the reports provided by APS.

APS provides a set of reports that help you understand your application and its various components. You can use these reports as you develop an application to determine the status of your work and the tasks left to complete. Some reports help you to troubleshoot problems in an application that you are developing, or to determine the impact of a proposed change. Others help you to verify the results of your work. Once you have fully implemented an application, use the APS reports to document it so that developers who later maintain or enhance the application can easily understand it in detail.

One of the reports most-used by developers is the Program DB/DC Report, which documents the screens and subschemas used by a program.

The Program DB/DC Report has three sections:

- Database views, showing any of the following applicable to your database target:
 - Record names of the root and each dependent segment

- Segments in hierarchical sequence, indented to show each segment's level in the data base
- Functions (PROCOPTS) that can be performed for each segment
- Key and sequence fields for accessing each record
- VSAM file descriptions
- Record I/O areas, showing the COBOL I/O areas for the database and file records
- Screen I/O areas, showing the COBOL I/O areas for up to 20 specified screens

Procedure Start this procedure where the last one ended - in the APS Main Menu.

- 1 On the Main Menu, type **2** in the **Option** field and press Enter. The Dictionary Services screen displays.

Figure 8-1. Access the Documentation Facility

```

----- APS Documentation Facility -----
OPTION ==>

  1 - Macro Cross Reference
  2 - Entity Cross Reference
  3 - Component List Report
  4 - Entity Content Report
  5 - Program DB/DC Report
  6 - Field/Screen Cross Reference Report
  7 - Entity Search Utility
  8 - Entity Parts List
  9 - Entity Use Report
 10 - DDIFILE Report (Submits Job)

```

- 2 To select Actions Documentation Facility, type **2** in the **Option** field and press Enter. The Documentation Facility screen displays, where you select the report to print.

Figure 8-2. Select Reports that Document the Application

```

OPTION ==> 5_
1 - Macro Cross Reference
2 - Entity Cross Reference
3 - Component List Report
4 - Entity Content Report
5 - Program DB/DC Report
6 - Field/Screen Cross Reference Report
7 - Entity Search Utility
8 - Entity Parts List
9 - Entity Use Report
10 - DDIFILE Report (Submits Job)

```

- 3 To select the Program DB/DC Report, type **5** in the **Option** field and press Enter. The report selection screen displays, where you enter the selection criteria.

Figure 8-3. Request the Report for the Program Subschema

```

COMMAND ==>
          PROGRAM ==> apsupd
          PSB/SUBSCHEMA ==> apsupd_
(optional) SCREEN(s) 1 ==>          2 ==>
                    3 ==>          4 ==>
                    5 ==>          6 ==>
                    7 ==>          8 ==>
                    9 ==>         10 ==>
                   11 ==>         12 ==>
                   13 ==>         14 ==>
                   15 ==>         16 ==>
                   17 ==>         18 ==>
                   19 ==>         20 ==>

ENTER to Submit Job or END to EXIT

```

- 4 Type **apsupd** in the **Program** field.
- 5 Type **apsupd** in the **PSB/Subschema** field.
- 6 Press Enter to execute the report and submit a batch job that prints it.
- 7 After the report has executed, you can optionally view it online. To do so, type the **tso sdsf IBM** command in the **Command** field and press Enter.
- 8 Press F3 repeatedly until you return to APS Main Menu.

Checkpoint The VSAM file report is generated. You can now define the program specifications in Online Express.

Hints APS provides a number of other reports describing application entities such as applications, screens, data structures, programs, scenarios, and report mock-ups. Through criteria you select, you can see the reports in a number of categories and formats.

9 Defining Program Specifications

In this chapter, you learn about Online Express, define program functions, and map screen fields.

Learn About Online Express

Goal In this section you learn the concepts of processing logic in Online Express, and you access Online Express to specify the Parts Update program logic.

You complete your application by defining its processing logic using Online Express, a menu-driven painter that offers a fill-in-the-blanks approach. You use Online Express for any type of online application. Online Express references the information that you have specified in the other APS painters and importers, and prompts you to define the processing logic for those specifications. You do the following to define the program in Online Express:

- Select predefined program functions. Online Express provides predefined program function logic, including teleprocessing and database read and write functions. You simply select the program function codes that you want.
- Define custom program functions. You can define your own program functions to supplement the predefined functions. End users can execute custom functions just as they execute any predefined function.
- Specify methods for executing functions. You specify the method by which the end user executes the functions. For example, the end user can either enter a code in a function field or press a key.
- Map screen fields to database fields. Online Express automatically displays all screen fields that you have defined in the Screen

Painter, so that you can map them to the appropriate fields in your database.

Procedure Start this procedure where the last one ended - in the Main Menu.

- 1 To transfer to the Painter Menu, type **1** in the **Option** field and press Enter. The APS Painter Menu displays.
- 2 Type **e** in the **Command** field to edit the application.
- 3 If necessary, type **ap** in the **Type** field to indicate Application Painter. To select the tutorial application, type the application name, **apsappl**, in the **Member** field and press Enter. The APS Application Painter screen displays.

Figure 9-1. Access Online Express

```

EDIT --- APPLICATION: APSAPPL ----- COLUMNS 000 0
COMMAND ==>                               SCROLL ==> CSA

      DC ==> ISPF          AUTHOR ==> USERNAME
      DB ==> USAM        SCREEN SIZE ==> MOD2

-----
- LINE-  PROGRAMS  SCREENS  IO  REPORTS  DATA STR TY SBSC/PSB USERMACS L
-----
000001  APSMNU     APSM     IO
000002  APSADD     APSA     IO          APSADD
000003  ox APSUPD     APSU     IO          APSUPD
**END**
  
```

- 4 To access Online Express, type **ox** in the selection field to the left of the APSUPD program and press Enter. The Online Express menu screen displays.
- 5 To access the Program Definition screen, type **1** in the **Command** field and press Enter.

Figure 9-2. Online Express Menu

```

COMMAND ==>

PROGRAM: APSUPD      SCREEN: APSU      SUBSCHEMA: APSUPD  DC TARGET: ISPF

1  PROGRAM DEFINITION - Specify program information and functions
2  ALTERNATE FUNCTIONS - Define application and IP functions
3  PF KEY FUNCTIONS   - Assign PF key functions
4  FIELD MAPPING      - Map screen fields to program fields
5  CONTROL POINTS     - Add application specific logic
6  DATA BASE ACCESS  - Specify data base access
7  SAVEKEY DEFINITION - Specify SAVEKEY storage requirements
SC APS SCREEN PAINTER - Invoke APS Screen Painter
P  EXPRESS PARAMS     - Specify Express Params

COMMANDS:  SAVE - COPY <name> - GEN - CAN - AUTO - REPORT - OKIN

```

Checkpoint You are in Online Express and are ready to begin specifying program logic by selecting the predefined function codes the program requires.

Help Online Express has an extensive help facility that you can select from the action bar, or access from your screen by pressing PF1.

When you access the Online Express Menu and the warning message DEFAULT PROGRAM NOT FOUND displays, press PF3 to erase the message from the screen.

Hints Other functions you can perform in Online Express are:

- Test the teleprocessing functions. Without accessing a database, you can execute your teleprocessing functions to verify that they work as desired. In addition, you can enter test data to verify that your field edits are properly defined.
- Customize the predefined functions. You can modify and supplement the default processing logic of the predefined functions by adding your own logic at predefined locations in your program, called control points.

Define Program Functions

Goal In this section, you specify basic program information and select predefined functions, and then specify the method for executing the functions.

This program:

- Reads database records and move data to screen fields - the Query function.
- Updates the records by moving data from screen fields to the records - the Update function.
- Deletes database records - the Delete function.
- Clears all screen fields of data - the Clear function.
- Terminates the program - the Exit function.

The end user will execute these database and teleprocessing functions by entering a function code in the screen field named Function. Additionally, the end user will press the PF3 function key to return control to the Main Menu program, which displays the application Main Menu screen.

Procedure Begin this procedure where the last one ended - in the Program Definition screen.

Figure 9-3. Select Predefined Program and Database Functions

```

COMMAND ==>

FUNCTION FIELD ==> FUNCTION           (Field where user enters func code)
FUNCTION CODES ==> qudce
(Valid Codes: Q=Query U=Update A=Add D=Delete N=Next F=Forward B=Backward
C=Clear E=Exit)

ROW FUNCTION FIELD ==>                (Field where user enters row code)
ROW FUNCTION CODES ==>
(Valid row codes: U=Update A=Add D=Delete)

SYSMSG FIELD ==> SYSMSG               (Field where messages are displayed)

```

- 1 To select the Query, Update, Delete database functions, and the Clear and Exit teleprocessing functions, type **qudce** in the **Function Codes** field, as shown in *Figure 9-3*.
- 2 To specify that the end user will press a function key to transfer between application screens, type **3** in the **Command** field and press Enter. The PF Key Functions screen displays.

Figure 9-4. Assign Functions to PF Keys

Function	Reserved Function or Function Name
PFKEY01	
PFKEY02	
PFKEY03	x apsmnu_
PFKEY04	
PFKEY05	
PFKEY06	
PFKEY07	
PFKEY08	
PFKEY09	
PFKEY10	
PFKEY11	
PFKEY12	

Functions: *=Reserved, P=Perform, G=Global code, L=Local code (E to edit), S=Send, X=Xactl, M=Msg-sw, C=Call, \$=Invoke macro

- 3 To specify that the F3 key returns control to the Main Menu program, do the following:
 - On the PFKEY03 input line, type **x** under the **Function** column.
 - Type **apsmnu** under the **Reserved Function or Function Name** column.
- 4 Optionally, you can duplicate the actions for PF15 so your end user can use either row of function keys on the terminal. To do so, type ***** in the **Command** field and press Enter. Optionally, press Enter again to display function keys PF13 to PF24; the actions and functions are duplicated, with X assigned to PF15.

Checkpoint You have defined your program specifications and specified how they are executed. Next, you map the screen fields to the appropriate fields in the imported VSAM database.

Help Teleprocessing (TP) functions transfer screen data and program control from the current program to another screen or program. The APS predefined TP functions include the following:

TP Function	Description
S(end)	Transmit an input/output screen.
M(SG-SW)	Schedule a new program and optionally pass a screen record or other data record to it.
X(CTL)	Transfer control to another program.
C(all)	Call a subroutine or perform a CICS LINK.
C(lear)	Move spaces to all I/O fields.
E(xit)	Terminate the program.

Database functions read from and write to your application's database. The APS predefined database functions include the following:

Read Function	Description
Q(uey)	Obtain one or more records and display data on the screen.
B(ackward)	Page backward through a repeated record block.
F(orward)	Page forward through a repeated record block.
N(ext)	Retrieve the next sequential record and display data; not applicable to SQL.
R(efresh)	Re-read the database when the end user executes any database write function on one or more repeated record block rows, and re-display the record block to reflect the database updates.
A(dd)	Store records.
D(elete)	Erase records.
U(pdate)	Modify records.

The screen design dictates which functions act on which fields. The Parts Update screen displays data one occurrence at a time, displaying information about a particular part. To query the record in this tutorial, the end user will enter **q(uey)** in the **Function** field and a value in the record key field, Part Number - you assigned the query function in step 1. Then, in step 3, you assigned the **X(CTL)** function to the F3 key so that the end user can press F3 to transfer to the Main Menu.

Any function that you define for this program acts on all fields. You can select any of the following functions:

A(dd)	D(elete)	N(ext)	S(end)
C(all)	E(xit)	Q(uey)	U(pdate)
C(lear)	M(SG-SW)	R(efresh)	X(CTL)

Hints The function field on the Parts Update screen has the default name **FUNCTION**. If you assign the function field a different name, you would enter that name in the **Function** field on the Program Definition screen.

The system message field on the Parts Update screen has the default name **SYSMSG**. If you assign the system message field a different name, you would enter that name in the **SYSMSG** field on the Program Definition screen.

Map Screen Fields

Goal In this section, you map the screen fields to the appropriate fields in the VSAM database. You specify each screen field's corresponding database field, and indicate whether the screen field is an input field, an output field, or both.

To help you do so quickly, Online Express displays all screen fields that you defined in the Screen Painter.

Procedure Start this procedure where the last one ended - on the PF Key Functions screen.

- 1 Type **4** in the **Command** field and press Enter. The Field Mapping screen displays, listing the screen fields you defined in the Screen Painter, prefixed by the screen name.

Figure 9-5. Initial Field Mapping Screen Lists Screen Fields

```

COMMAND ==> *
  Screen Field          I/O/B   Program Field
-----
  APSU-FUNCTION
  APSU-PART-NBR
  APSU-SHORT-DESC
  APSU-UNITS-INSTOCK
  APSU-BASE-PRICE
  APSU-DIMENSIONS
  APSU-SAVEKEY
  APSU-SYSMSG

Line cmds: E=Field edits, I=Insert, D=Delete (Continuation Lines Only)

```

- 2 To enter the names of the database fields corresponding to each screen field and indicate that field data moves both from the screen to the database record and vice versa, type * (asterisk) in the **Command** field and press Enter. APS copies the field names from the **Screen Field** to **Program Field** for you and enters the default of **B** in the **I/O/B** column.

Figure 9-6. Map Screen Fields with the * Command

```

COMMAND ==> prefix pm-
  Screen Field          I/O/B   Program Field
-----
  APSU-FUNCTION
  APSU-PART-NBR          B      PART-NBR
  APSU-SHORT-DESC       B      SHORT-DESC
  APSU-UNITS-INSTOCK    B      UNITS-INSTOCK
  APSU-BASE-PRICE       B      BASE-PRICE
  APSU-DIMENSIONS       B      DIMENSIONS
  APSU-SAVEKEY
  APSU-SYSMSG

```

- 3 To assign the database record prefix to the program field names, type prefix **pm-** in the **Command** field and press Enter. APS assigns the **PM-** prefix to the program field names.

Figure 9-7. Assign a Prefix to Program Field Names

COMMAND ===>		
Screen Field	I/O/B	Program Field
- APSU-FUNCTION		
- APSU-PART-NBR	B	PM-PART-NBR
APSU-SHORT-DESC	B	PM-SHORT-DESC
APSU-UNITS-INSTOCK	B	PM-UNITS-INSTOCK
APSU-BASE-PRICE	B	PM-BASE-PRICE
APSU-DIMENSIONS	B	PM-DIMENSIONS
APSU-SAVEKEY		
APSU-SYSMSG		

Checkpoint You have completed field mapping and are ready to define program database access.

Help Use the prefix command in any of the following formats:

pre[fix] fldprefix	Add fldprefix to fields on all lines.
pre[fix] fldprefix m n	Add fldprefix to fields from line m through line n .
pre[fix] fldprefix * n	Add fldprefix to fields from line 1 through line n .
pre[fix] fldprefix n *	Add fldprefix to fields from line n through the last line.

Notice that the FUNCTION, SAVEKEY, and MESSAGE fields were not copied, because they don't correspond to database record fields - they're defined only in your program and are reserved for special purposes in Online Express.

Hints Use of the * command assumes the screen field names are the same as the database record names (except for the prefixes), which is true for this program.

To clear some or all of the values you entered, you can enter the reset command in any of the following formats in the **Command** field:

reset	Clear values on all lines.
reset m n	Clear values from line m through line n .
reset * n	Clear values from line 1 through line n .
reset n *	Clear values from line n through the last line.

10 Defining Database Access

In this chapter, you learn about database access through Online Express, and define and customize database accesses.

Learn Database Access through Online Express

Goal In this section, you learn about accessing databases through Online Express, and transfer to the screen where you begin defining the database access specifications.

You do the following to define database access in Online Express:

- Define database access calls. For each database function that you select, you define one or more database calls that specify which record or records to read, and which database actions to perform on them, such as obtain, modify, store, and erase.
- Customize the predefined functions. You can modify and supplement the default processing logic of the predefined functions by overriding the default error processing of database calls.

An Online Express database call defines which record or records to read, and which database actions to perform on them. Each database function that you specify in your program definition has a corresponding database action that defines the function, as shown below:

Database Function	Corresponding Action
Query	Obtain
Update	Modify
Add	Store
Delete	Erase

You define a database call by completing a few Online Express screens that prompt you to do the following:

- Select which record or records to access.
- Specify the database read and write actions that you want to perform on the record.
- Qualify the data that you access by specifying field criteria.

Procedure Start this procedure where the last one ended - on the Field Mapping screen.

- 1 Type **6** in the **Command** field and press Enter. The Database Access Summary screen displays. The first time you see this screen it is blank, except for call numbers. From here you access other screens to define the database call specifications.

Figure 10-1. Begin Database Access Specifications

Function	Action	Data Base Record	Qualifier	Nesting	B
01	DME	PART-MASTER-REC	*REQUAL	0	
02					
03					
04					
05					
06					
07					
08					
09					
10					
11					
12					
13					
14					
15					

S=Select Q=Qualification I=Tailoring M=Move A=After D=Before I=Insert D=Delet

- 2 To specify database access for the first call, type **s** in the selection field to the left of the first call, **01**, and press Enter. The Database Record Selection displays.

Checkpoint You are ready to define the access methods for the database calls.

Help You should define the calls in the order in which you want to execute them, but you can rearrange the order and modify any call definition at any time.

Hints Other functions you can perform in Online Express are:

- Define savekey storage. You define savekey storage area(s) to store record key values during program execution if your program does any of the following:
 - Update records with the U(pdate) and D(elete) program functions.
 - Obtain records sequentially with the N(ext) program function
 - Display repeated record blocks that the end user can scroll with the F(orward) and B(ackward) functions.

Note that you already defined the savekey in the Screen Painter. You can define it in either place.

- Define Commarea storage. You can define an area in your program to store any data that your program passes between programs, called a Commarea. You must define a Commarea if your program passes data with the X(CTL), M(SG-SW), or C(all) functions. You do so simply by specifying its size to Online Express.

Define the Accesses

Goal In this section, you specify the database accesses, and then you define selection criteria for obtaining the record.

The Parts Update program queries, updates, and deletes a record, so you must define a call to obtain, modify, and erase that record. You do so by entering an action code next to the record that Online Express displays on the Database Record Selection screen.

Then, on the Database Qualification Selection screen, you define selection criteria for the Obtain access by entering an operator and a qualification value next to the field. In the Parts Update program, the part number of the record to be retrieved from the database must equal to the number entered by the end user in the PART-NBR field on the Parts Update screen.

Procedure Start this procedure where the last one ended - on the Database Record Selection screen.

Figure 10-2. Specify the Database Accesses

CALL NUMBER: 01 ----- DATA BASE RECORD SELECTION -----			
COMMAND ==>			
Action	Data Base Record	Sequence Field/Set Name	Type
OME	s PART-MASTER-REC	PH-PART-NBR	USAM

- 1 To assign the obtain, modify, and erase functions, type **ome** in the **Action** field to the left of PART-MASTER-REC, as shown in *Figure 10-2*.
- 2 To qualify the Obtain access, type **s** in the selection field to the left of PART-MASTER-REC and press Enter. The Database Qualification Selection screen displays.

Figure 10-3. Qualify the Obtain Access

CALL NUMBER: 01 ----- DATA BASE QUALIFICATION -----				
COMMAND ==>				
Field Name	Ty Op	Value	Len	Bool
RECORD: PART-MASTER-REC		ACTIONS: OME		
PH-PART-NBR	PR =	APSU-PART-NBR	00008	

- 3 To specify that the part number of the record to be retrieved must equal to the number entered by the end user in the PART-NBR field on the APS Update screen, type **=** in the **Op(erator)** field and **apsu-part-nbr** in the **Value** field.
- 4 Press F3 to save your specifications. Press F3 again. The Database Access Summary screen displays, showing the database access, or call, defined for this program. This program can Obtain, Modify, and Erase (as indicated by O, M, and E in the Actions field) PART-MASTER-REC records, and obtain a customer record qualified on a key entered by the end user (*KEYQUAL).

Figure 10-4. View Your Database Calls

```

COMMAND ==>
-----
Function Action      Data Base Record      Qualifier  Nesting  B
-----
t 01          QME      PART-MASTER-REC      *KEYQUAL  0
02
03
04
05
06
07
08
09
10
11
12
13
14
15
S=Select Q=Qualification T=Tailoring M=Move A=After B=Before I=Insert D=Delet

```

Checkpoint You have defined the database accesses for the Parts Update program. Now you need to define custom logic for error handling.

Help On the Database Access Summary screen, you can rearrange, add to, and delete from the list as follows:

- Move any call before or after another call by typing **m** (move) next to the call, and either **b** (before) or **a** (after) next to another call.
- Add a call definition in between calls in the list by typing **i** (insert) next to a call and then defining the new call.
- Delete a call by typing **d** (delete) next to the call.

Hints A qualification value can be a COBOL screen field, a Working-Storage name, or a literal enclosed in quotation marks. You can qualify the following types of fields, as shown below. Each field's type automatically displays in the **Ty(pe)** field on the screen.

Field Type	Description
KY	Key field. To qualify on a partial key, type over the value in the Len(gth) field.
PR	Primary index.
SR	Non-unique search field.

You can qualify data based on your database target, as follows:

Database	Qualification Method
IMS	Qualify calls on any field, including: <ul style="list-style-type: none"> • Key field • Non-unique search field • Sequence field Qualify calls on multiple fields and conditions using Boolean operators.
SQL	Qualify calls on any column. Qualify calls on multiple columns and conditions using Boolean operators. Qualify calls on multiple columns of multiple tables, using Union and Join calls. Qualify calls with Subselect specifications, including SQL column and scalar functions, and Exists, Group By, and Having clauses.
VSAM	Qualify calls on any field, including: <ul style="list-style-type: none"> • Primary index • Partial key field
IDMS	Qualify calls on any field, including: <ul style="list-style-type: none"> • Address • CALC key • Key • Non-unique search • Sequence

Customize Database Access

Goal In this section, you customize database access by overriding the default status code and error messages, and, by inserting obtain logic at a database access control point.

You can customize database call processing to override the status codes of Online Express status flags and the text of default error messages on the Database Call Tailoring screen.

Online Express provides five status flags. By default, all status flags except OK-ON-REC return the Error status code, as shown below:

Status Flag	Default Status Code
OK-ON-REC	N(ormal)
END-ON-REC	N(ormal)
NTF-ON-REC	E(rror)
DUP-ON-REC	E(rror)
VIO-ON-REC	E(rror)

When Online Express returns the Error status flag, the program aborts and performs the ERROR-SEND-AND-QUIT paragraph.

To override the default Error flag processing, you can change a status flag's status code from Error to Exception, and then write your own error routines at control points on the Call Tailoring screen.

To write your own error routine, you code a group of Procedure Division statements, known as a program stub, in the Specification Editor.

Procedure Start this procedure where the last one ended - on the Database Access Summary screen.

- 1 Type **t** in the selection field for call **01**. The Database Call Tailoring screen displays.
- 2 To specify the Obtain access, type **o** in the **Action to be Tailored** field.
- 3 To override the standard error message, type **part number entered not found** in the **Error Message** field.

- To indicate that the error message is text, type **t** in the **Error Message Type** field.

Figure 10-5. Customize Database Calls

```

COMMAND ==>
RECORD NAME: PART-MASTER-REC
ACTION TO BE TAILORED ==> o ACTIONS SPECIFIED: OME
Control Points      Action      Control Point Name
-----
BEFORE DB ACCESS
NORMAL STATUS
EXCEPTION STATUS
ERROR STATUS        e
AFTER DB ACCESS

Actions: $=Macro call, P=Perform, G=Global code, L=Local code (E to edit)
STATUS MATRIX  OK ==> N  END ==> N  NTF ==> E  DUP ==> E  UID ==>
(N=Normal, X=Exception, E=Error)

ERROR MESSAGE:    ==> part number entered not found
ERROR MESSAGE TYPE ==> t {S=Standard I=Text M=Macro}
ABORT ON ERROR    ==> Y {Y or N}

```

- To insert custom logic, type **e** in the **Action** field for the Error Status control point and press Enter. The Specification Editor displays, where you paint a program stub.

Figure 10-6. Write Program Code

```

EDIT --- PROGRAM: APSUPD CP MEMBER: CXOER01 ----- COLUMNS 001 0
COMMAND ==> SCROLL ==> PAG
-LINE- -KVWD- 12-----20-----30-----40-----50-----60-----7
***** TOP OF DATA *****
000100 MOVE SPACES TO APSU-SHORT-DESC-INPT
000200 MOVE SPACES TO APSU-UNITS-INSTOCK-INPT
000300 MOVE SPACES TO APSU-BASE-PRICE-INPT
000400 MOVE SPACES TO APSU-DIMENSIONS-INPT
***** BOTTOM OF DATA *****

```

- To initialize the record fields, type the code as shown in *Figure 10-6* above and press Enter. Check that you typed the code correctly.
- Press F3. The Database Call Tailoring screen displays.
- To specify the Modify access for modification, type **m** in the **Action to be Tailored** field and press Enter. The information for the Modify access displays.
- Type **part number entered not found** in the Error Message field to override the standard error message. Next, type **t** in the **Error**

Message Type field to indicate that the error message is text. Then press Enter.

- 10 To specify the Erase access for modification, type **e** in the **Action to be Tailored** field and press Enter. The information for the Erase access displays.
- 11 Type **part number entered not found** in the **Error Message** field to override the standard error message. Next, type **t** in the **Error Message Type** field to indicate that the error message is text. Then press Enter.
- 12 Press F3 twice to save your work and return to the Online Express Menu. Press F3 again to return to the Application Painter.

Checkpoint You have completed all the development work for the Parts Update program and screen, and are ready to set some generation options and generate the application.

Help Control points are locations in the generated program where you can insert your own logic, in the form of program stubs.

A program stub can be local, applying only to the current program, or global, applying to any program in the application. You code local stubs in the Online Express Specification Editor, as you did in this procedure; you code global stubs in the Program Painter.

The Status control points on Database Call Tailoring let you insert a program stub to alter the standard error processing. This logic executes after the database access.

Hints You can prevent the Error flag from aborting the program by specifying **n** for the **Abort On Error** field on the Database Call Tailoring screen.

11 Generating the Application

In this chapter, you learn about the generation process and view generated output.

Learn about the Generation Process

Goal In this section, you learn about the APS Generator and APS Precompiler and the tasks they perform. You also generate the application.

When you generate an application, the APS Generator and APS Precompiler translate your APS specifications into a complete, structured COBOL application. APS then automatically passes the source to your COBOL compiler and link edit program to produce a load module.

When you generate an application, APS automatically processes your application as follows:

- Ensures that each component of your application exists.
- Generates APS screen symbols for each screen, for use by the APS Precompiler.
- Generates the screen source, native to your DC environment, such as BMS or MFS mapsets.
- Arranges all APS program specifications into proper COBOL program organization.
- Includes all the externally-defined components - such as APS macros, user-defined Customization Facility macros, COBOL copylibs, and APS data structures - at the program locations you specify in the Application Painter, Specification Editor, and Program Painter.
- Processes all APS database and data communications calls and user-defined macros, translating all source into COBOL source.

- Writes a temporary error message file and sorts it into the COBOL compiler error message file. The combined error message file presents messages sorted by program line number, with both types of messages appearing where appropriate.

The generation process produces a separate output file for each screen and COBOL program selected. The screen source is generated first, followed by each program in the order in which they appear on Application Painter. APS stores your generated and precompiled COBOL program source and screen in your user Project and Group.

Procedure Start this procedure where the last one ended - in the Application Painter.

- 1 To tailor screen generation to your target environment, do the following:
 - a Transfer to the Screen Painter by typing **s** in the selection field to the left of the APSU screen name. Press Enter.
 - b Type **pm** on the Command line and press Enter. The Screen Generation Parameters screen displays.
 - c Because the screen name is not the same as the program name, you must specify the program name. Type **apsupd** in the **Associated Pgm** field.
 - d Press PF3 repeatedly until you return to the Application Painter.
- 2 Optional. Before generating, you might want to verify that your generation options DC and DB targets are set to ISPF and VSAM respectively. To do so, type **OPT 1** in the **Command** field and press Enter to access the APS Generator Options panel.
- 3 To generate the application, type **gen** in the **Command** field and press Enter, as shown in *Figure 11-1*.

Figure 11-1. Generate the Application

```

EDIT --- APPLICATION: APSAPPL ----- COLUMNS 000 0
COMMAND ==> gen_                      SCROLL ==> CSR

      DC ==> ISPF
      DB ==> USAM                        AUTHOR ==> THERESAG
                                         SCREEN SIZE ==> MOD2

-----
-LINE-  PROGRAMS  SCREENS  IO  REPORTS  DATA STR TY SBSC/PSB USERMACS L
-----
000001  APSMNU     APSM     IO
000002  APSADD     APSA     IO
000003  APSUPD     APSU     IO
                                         APSADD
                                         APSUPD
**END**

```

Checkpoint You have generated the application and should review the system messages.

Help The generators first process the screens serially, followed by each program.

You can tailor how APS generates your application. Options are available for controlling both the Generator and Precompiler, as well as target-specific options such as SQL Bind and IDMS options.

Hints To observe the generation process, go to your system area, just as you would for any job you submit.

View Generated Output

Goal In this section, you review the system message files in the generated output for errors, and then delete the output to save disk space.

You can view the system messages for the generated screens online. These output files contain the equivalent of MVS JCL messages as well as the specific APS runtime messages.

To save space, you should delete the generated output files. APS will flag each output file as deleted so that these jobs no longer appear in the jobqueue.

- Procedure**
- 1 To view the output files, type **tso sdsf** in the **Command** field and press Enter.
 - 2 Type **s** to the left of each output file and press Enter. The first output file displays.
 - 3 Press PF7 and PF8 to scroll throughout the first output file.
 - 4 Press PF3 to view the next selected output file.
 - 5 Continue reviewing your output using PF7, PF8, and PF3 until you return to the output jobqueue screen.
 - 6 To save space, type **d** to the left of each job name and press Enter. The files will be deleted.
 - 7 Press F3 twice to return to APS Main Menu.

Checkpoint You are now ready to execute, or run, the entire application.

12 Running the Application

Goal In this section, you run your application from the APS Prototype Execution facility and test the application functionality.

The APS Prototype Execution facility lets you run a single program or the entire application. For this application, if you run the first program, the menu program, it automatically calls the others.

Procedure Start this procedure where the last one ended - in the APS Main Menu.

- 1 To access the Prototype Execution Facility, type **3** in the **Option** field and press Enter. The Prototype Execution screen displays.

Figure 12-1. Select the Prototype Execution Facility

```

----- APS Main Menu -----
OPTION ==>

0 - Options
1 - Painters
2 - Dictionary Services
3 - Prototype Execution
4 - Utilities
5 - Version Control System

C - Changes in APS for z/OS
T - APS Tutorial
  
```

- 2 To run the application, type **1** in the **Option** field and type **apsmnu** in the **Program** field. Type **YES** in the **Datasets to be allocated by CLIST()** field. Then press Enter.

Figure 12-2. Run the Application

```
----- APS Prototype Execution -----
OPTION ==>

  1 - Run program from LOADISPF
  2 - Run application from LOADISPF
  3 - Run application from LOADIMS via BTS

  APPLICATION ==>
    or PROGRAM ==>

  Datasets to be allocated by CLIST()           ==>
  DB2 (Run from LOADISPF via DSN)              ==>
  Verify load modules prior to execution       ==>
```

Checkpoint The application is executed. Exercise the application by adding and inquiring on various part numbers.

Help You can exit the application at any time by pressing F3. To exit APS, continue pressing F3.

Hints For further information on the steps you performed in this tutorial, see your User's Guide.

Index

A

- Application Painter
 - accessing 17
 - accessing other painters from 20
 - purpose of 19
- application, sample 7
- applications
 - application definition, creating 19
 - components of, defining using APS 19
 - executing 79
 - generating 75
- APS, starting 13
- attributes, field 26
 - assigning to a specific field 27

C

- Caps option, Screen Painter 23
- Commarea, defining 67
- comments, writing
 - in Scenario Painter 31
- control points
 - database call 71
- conversion values, field edits 45
- copying
 - field edits on character screens 48
- customizing
 - Online Express programs, database call
 - error processing 71

D

- data simulation in Scenario Painter 32
- Database Access Summary screen 66
- Database Call Tailoring screen 71
- database calls, Online Express
 - customizing 71
- database functions
 - Online Express 65
- database functions, Online Express
 - customizing 71
- database importers
 - VSAM 49
- Database Qualification Selection screen 68
- Database Record Selection screen 67
- DDI statements
 - VSAM 49
- DDISYMB, generating
 - VSAM 49
- deleting
 - character screen fields and text 26
 - field edits on character screens 47

E

- edits, field, error handling 39, 48
- edits, fields on character screens
 - conversion values for 45
 - copying 48
 - deleting 47
 - input 41
 - internal picture 42
 - output 44

- overview of 37, 39
- user-defined 39
- value ranges for 45
- error handling
 - database calls, Online Express, customizing 71
 - field edits 39, 48
- executing applications 79

F

- Field Mapping screen 61
- Field Selection screen, field edits 39
- function codes, Online Express program 58
- function fields, Online Express program 58
- functions, Online Express program 58

G

- generating applications 75
- generation parameters
 - screen 76

I

- I/O fields
 - creating 24
- importing databases
 - VSAM 49
- input field edits
 - for character screens 41
- internal picture field edits
 - for character screens 42

K

- Keys option, Screen Painter 23

M

- Main Menu screen 15
- mapping, field 61

N

- Nulls option, Screen Painter 23

O

- Online Express
 - control points, database call 71
 - customizing programs 71
 - database functions 65
 - mapping fields 61
 - overview of 55
 - processing logic, defining 58
 - teleprocessing functions, specifying for
 - character programs 58
- Online Express menu 56
- Options Menu screen 16
- output field edits
 - for character screens 44

P

- Painter Menu screen 17
- PF Key Functions screen 58
- PF keys
 - assigning program functions to 58

- Profile option, Screen Painter 22
- Program DB/DC Report (PG02) 50
- Program Definition screen 58
- programs
 - specifying in application definition 20
- Project Group Environment screen 16
- Prototype Execution screen 79
- prototyping under ISPF 7
 - field names 28

R

- reports, APS
 - Program DB/DC Report (PG02) 50
- RETITLE command
 - Scenario Painter 30
- RUN command
 - Scenario Painter 31

S

- sample application 7
- savekey storage, defining 24
- Scenario Painter
 - see scenario prototype
 - overview of 29
- scenario prototype 29
 - accessing Scenario Painter 29
 - creating and running 29
 - data, passing 32
 - screen sequence, defining 30
 - screen sequence, modifying 35
 - screen titles for, assigning 24
- screen fields
 - deleting 26
 - I/O fields, creating 21, 24
 - length, changing 26
 - naming conventions 28
 - text fields, creating 21, 24
- screen generation parameters 76

- Screen Painter
 - overview of 21
 - user profile 22
- screens
 - data, passing in scenario prototype 32
 - layout, designing 22, 24
 - scenario of sequence, creating and running 29
 - specifying in application definition 20
 - titles, assigning for Scenario Painter 24, 25
- screens, APS
 - Application Painter 19
 - Database Access Summary 66
 - Database Call Tailoring 71
 - Database Qualification Selection 68
 - Database Record Selection 67
 - Field Mapping 61
 - Field Selection 39
 - Main Menu 15
 - Online Express menu 56
 - Options Menu 16
 - Painter Menu 17
 - PF Key Functions 58
 - Program Definition 58
 - prototype execution 79
 - Scenario Painter 29
 - Screen Generation Parameters 76
 - Specification Editor 72, 75
- Specification Editor screen 72
- starting APS 13
- subschemas
 - specifying in application definition 20
- system messages
 - for character screens, creating field for 24

T

- teleprocessing functions, Online Express
 - specifying for character program 58
- text fields
 - creating 24

TITLE command, Screen Painter
effect in scenario prototype 30

U

user-defined field edits 39

V

value ranges, field edits 45

VSAM

files, importing 50