IBM

Screen Definition Facility II

Preparing a Prototype

Release 3

IBM

Screen Definition Facility II

Preparing a Prototype

Release 3

```
┌─ Note! ──────────────────────────────────────────────────────────────────┐
│                                                                            │
│  Before using this information and the product it supports, be sure to read the general information under │
│  "Notices" on page v.                                                      │
│                                                                            │
└────────────────────────────────────────────────────────────────────────────┘
```

Contents

Notices

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

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

Trademarks and service marks

The following terms, denoted by an asterisk (*), used in this publication, are trademarks or service marks of IBM Corporation in the United States or other countries:

IBM
MVS/ESA
MVS/XA
VM/ESA
VM/XA

Before starting

You should work through the *SDF II General Introduction* before you start to read this book. The *SDF II General Introduction* describes the definition of simple panels.

The SDF II prototype facility works with panels and rules defined by you. The panels are those which you define for your target system. The rules are statements that control the flow of panels. This book teaches you how to define the rules and run prototypes.

Instructions given in this book

The procedures in this book request that you communicate with SDF II in two different ways, namely:

- Enter
- Type

When you are requested to *Enter* data, a commonly used phrase is "Enter the high-lighted data," you must type in the characters shown in **highlighted** form and then press the Enter key.

When you are requested to *Type* data, then simply type in the characters shown in **highlighted** form. Instructions will follow as to what to do next.

Command types

There are two types of commands:

Panel commands entered on the command line

Line commands entered in the line command area of the screen.

Panel displays

The first time a new panel is referred to in the manual, it will be shown in full. It may be shown again in its full form if there have been significant changes made to it or if it has not been shown for many pages.

Most of the time, when a small number of fields of a panel are being updated, then only a small portion of that panel will be displayed and it will not be bounded by lines.

Getting help information

The following online help is available in SDF II:

Panel help
Field help
Messages and message help
Online reference.

Panel help

To get information about the SDF II panel you are using, press the Help key (PF1). SDF II displays the relevant online reference topic.

Field help

To get information about an entry field, enter a question mark (?) in the field. You then see a help panel that explains the purpose of the field and lists the possible entries. To get more information, you can then press the Enter key to view the online reference for the relevant topic.

Messages and message help

SDF II displays short messages in the upper right corner of the panel. Long messages are displayed on the line above the command line.

When SDF II displays a short message and you need more information, press the Help key (PF1) to get a longer message. To get more information, press the Help key (PF1) again. SDF II displays a help panel that explains the message more fully and describes the required user action and, if relevant, the system action.

For more information, from the help panel press the Enter key. SDF II displays the online reference for the relevant topic.

The online reference

The online reference contains how-to information and explains SDF II panels and commands.

You can start the online reference in these ways:

- Press Enter from an SDF II help panel.
- Press the Help key (PF1) from an SDF II panel.
- Select the online reference from the Select an SDF II Function panel.

What you will find inside

Chapter 1, "Prototype approaches" defines the term prototype. It describes the simulative and operational prototype approaches. How SDF II supports the two approaches is also described.

Chapter 2, "Prototype examples" describes prototype examples. It discusses three examples: two simulative and one operational.

Chapter 3, "Simulative prototype example" shows you how to define and run the two simulative forms of the prototype.

Chapter 4, "Operational prototype example" shows you how to define and run the operational form of the prototype.

Chapter 5, "Printing and library listing facilities" shows you how to print your prototypes and list the contents of the libraries.

The glossary explains terms and abbreviations used in this book that may be unfamiliar to you.

Summary of changes

The following changes were made to this book for SDF II Release 3:

- A new function, LIST, provides a facility for the user to list the prototypes held in libraries. The prototypes can be listed in the following sequences:

  - Prototype name
  - Library name
  - Modification date.

- The other SDF II prototype dialogs can be called from the List Prototypes Dialog.

- Usability has been improved by including the lengthy routines needed to run the operational prototype in the SDF II sample library.

The following changes were made to this book for SDF II Release 2:

- The Identify Prototype panel allows you to enter the name of an existing prototype to use as the skeleton of a new prototype.

- The panel command SAVE has been added.

- Rules can be continued on more than one line of the dialog.

- A new function allows you to print a prototype.

Summary of changes

Chapter 1. Prototype approaches

A prototype is an early model of an application, with limited function, which is used to test the application before the detailed design is started. Using a prototype approach can thus provide early feedback to application designers. This feedback may result in changes to the design before any application code has been written.

In application development there are two levels of prototypes:

- A simulative prototype
- An operational prototype.

The simulative prototype

The main purpose of a simulative prototype is to determine the initial requirements of application users before work begins on designing the application. A simulative prototype used early in the development cycle, and may be developed by end-users.

A simulative prototype shows the flow and layout of the panels used in an application. Because of this these prototypes are sometimes referred to as "screen prototypes." Simulative prototypes may also provide limited functions such as accepting data entries and displaying sample values. No coding is required.

The operational prototype

The purpose of an operational prototype is to validate the requirements of application users beyond the initial impression provided by a simulative prototype. Operational prototypes model the main functions of an application. For example, database access and limited error handling can be implemented in an operational prototype.

Operational prototypes require some coding and are therefore more expensive to build.

The SDF II prototype approach

SDF II provides all the functions required to define and run simulative and operational prototypes. Once you have created a simulative prototype, it can be expanded into an operational prototype by adding any functions expected in the application.

The SDF II prototype facility runs under ISPF. In an MVS environment, the prototype facility uses REXX EXECs or CLISTs to perform logical functions. In a VM environment, these functions are performed by REXX EXECs.

Defining a prototype

You define a prototype by following these steps:

1. Define the panels for the target system under which your application will run. These panels will be used in your prototype.

2. Generate the panels using APPLICATION PROTOTYPING as the target system.

> **Note:** The output library that you specify on the Specify Generation Parameters panel must be defined as a panel library when invoking the SDF II prototype function.

3. Enter the command **sdf2p** to invoke the prototype function. If that command does not work, find out from your system programmer how to invoke the SDF II prototype function.

4. Define the prototype rules in the Specify Prototype dialog. In their simplest form, these rules produce a simulative prototype for which no program code is required. Add your own routines, which can be either REXX EXECs or CLISTs, to expand a simulative prototype into an operational prototype.

   The panel flow can be controlled by function keys or by data input. If you wish, you may add initial values, sample values, or both. These values are defined when you create panel formats. Initial values, if defined, are shown when a panel is first displayed only if the variable has not been used before. If the variable has been used then its current value will be displayed. Sample values are shown only if you requested them in the rules you define for the prototype.

You are then ready to demonstrate the prototype to the application user.

Chapter 2. Prototype examples

This book describes three prototype examples. Each prototype represents a step in the design of an application. The example application, which uses four panels, is a system for entering customer orders. The same four panels are used in each of the three prototypes outlined below.

With **simulative prototype 1**, you display one panel after another in any order by entering the number of the panel on the command line. Any data you enter on a panel is not stored.

One of the panels is a selection panel. To return to the selection panel, press the End key (PF3). If you press the End key (PF3) again, you leave the prototype. SDF II then displays the Identify Prototype panel.

Simulative prototype 2 is similar to simulative prototype 1, but includes sample data. You can type over the sample data, but your changes are not stored.

Operational prototype 1 is the final version of the prototype. You enter data which is stored in a database. You can also update the data. The order list shows the contents of the database. The ability to store and retrieve data makes this an operational prototype.

A system for entering customer orders

The panels used in the prototypes are as follows:

Selection Panel	PROT00	From this panel you select any one of the following three panels or leave the prototype.
Order Entry	PROT10	Enter an order on this panel.
Order Update	PROT20	Make changes to the details of orders on this panel.
Order List	PROT30	This panel displays a list of orders that you have entered. You can scroll forward or backwards in this panel.

The flow of panels in the prototypes looks like this:
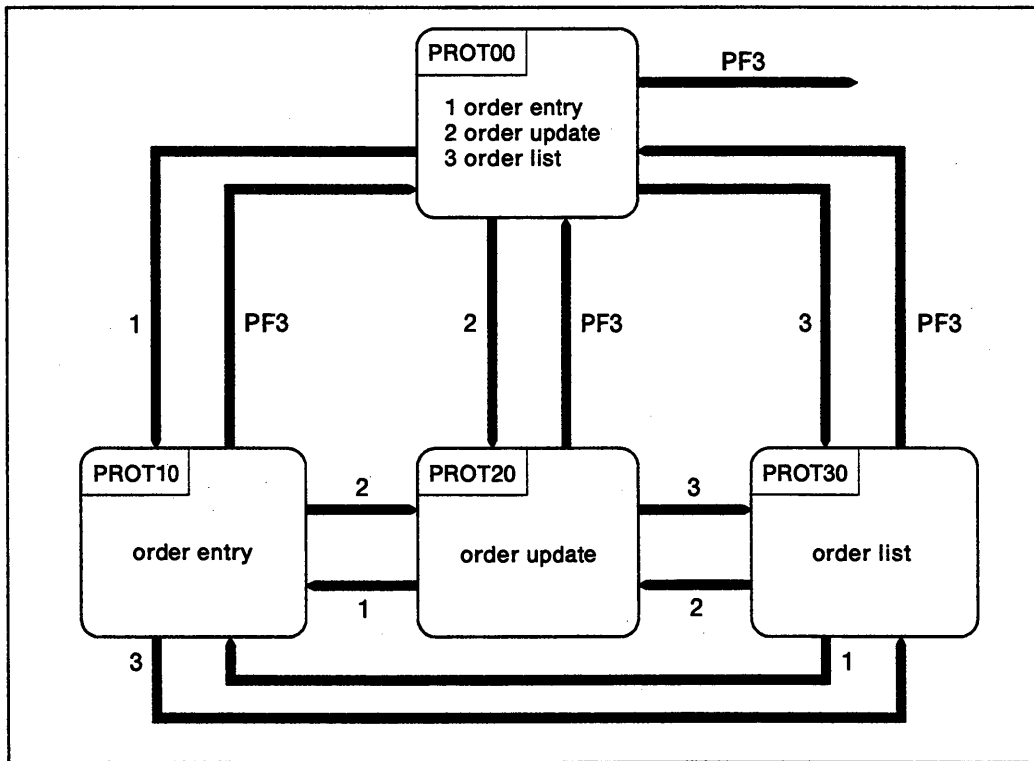
Define the panels

Pages 5 through 8 show the layout of the four panels. Define each panel as follows:

1. Select the target system for which you normally define panels.

2. Define the panel format to resemble the layout shown on each page.

3. Define the fields for each panel exactly as described.

If you have any problems, refer to the *SDF II General Introduction.*

In the sample panels on the following pages, variable fields are shown by under-scores. Fields are listed in the tables in the order they appear on the panel. Enter the panel text as it is shown in the sample panels.

In the sample panels the System message field should be defined as the last line of the panel.

Note: Do not start a variable field in column 1 of any row.

Selection panel (PROT00) layout

```
                          Order Entry and Query System
  ───────────────────────────────────────────────────────────────────

  Select one of the following functions

                      1  -  Order entry

                      2  -  Order update

                      3  -  List orders

  Press the End key (PF3) to terminate

  ==>  _
```

For panel PROT00, define the following:

Figure 1. Field definitions for panel PROT00

Field	Name	Protection	Length (chars)	Sample value
Command line	CMD	Unprotected	1	None
System message	MSG	Protected	79	None

Order entry panel (PROT10) layout

```
                              Order Entry
  _____

    Enter customer number
     Customer number            _____

    Enter item and quantity
     Item type                  _____
     Quantity                   ____

     Press the End key (PF3) to terminate

    ==>  _
```

For panel PROT10, define the following:

Figure 2. *Field definitions for panel PROT10*

Field	Name	Protection	Length (chars)	Sample value
Customer number	CUSTNO	Unprotected	6	MON12X
Item type	ITEM	Unprotected	12	WATCH
Quantity	QUAN	Unprotected	4	16
Command line	CMD	Unprotected	1	None
System message	MSG	Protected	79	None

Order update panel (PROT20) layout

```
                                Order Update


        _____


        Order to be modified
          Customer number          _____
          Item type                _____
          Quantity                 ____

        Modified content of order
          Item type                _____
          Quantity                 ____

        Press the End key (PF3) to terminate

        ==>  _
```

For the panel PROT20, define the following:

Figure 3. Field definitions for panel PROT20

Field	Name	Protection	Length (chars)	Sample value
Customer number	CUSTNO	Protected	6	MON12X
Item type	ITEM	Protected	12	WATCH
Quantity	QUAN	Protected	4	16
Modified item type	MITEM	Unprotected	12	PEN
Modified quantity	MQUAN	Unprotected	4	25
Command Line	CMD	Unprotected	1	None
System message	MSG	Protected	79	None

Order list panel (PROT30) layout

```
                              Order List
_____

   The following orders have been registered

   Currently displayed: Order ___ of ___

   Customer number     Item type        Quantity
   _____            _____        ____
   _____            _____        ____
   _____            _____        ____
   _____            _____        ____

   Press the End key (PF3) to terminate
   Scroll keys (PF7 backward, PF8 forward)

   ==>  _
```

For panel PROT30, define the following:

Figure 4. Field definitions for panel PROT30

Field	Name	Protection	Length (chars)	Sample value
Currently displayed order	RTP	Protected	3	3
Total number of orders stored	RTO	Protected	3	17
Customer number (n)*	CUSTNOn*	Protected	6	MON12X POW02A AVE12B ELD02X
Item type (n)*	ITEMn*	Protected	12	PEN WATCH SHOE BELT
Quantity (n)*	QUANn*	Protected	4	25 16 20 37
Command line	CMD	Unprotected	1	None
System message	MSG	Protected	79	None

Note: * n=1 through 4, according to the order in which each field appears on the panel

Generate the panels

After you complete the definition of the four panels, generate them by using the
APPLICATION PROTOTYPING option on the Identify Object for Generation panel.

Note: The output library that you specify on the Specify Generation Parameters
panel must be defined as a panel library when invoking the SDF II prototype
function.

Generate the panels

Chapter 3. Simulative prototype example

After defining the panels, establish the rules for the prototype. The rules determine the flow of control between the panels, according to entries made on the command line. In this chapter, you define the rules for the two simulative prototypes, PROT1 and PROT2.

MVS users: Check with your system programmer whether your installation uses REXX EXECs or CLISTs. You need this information to specify a prototype that is consistent with your installation's practice. Also ensure that the prototype invocation defines the prototype library.

VM users: Follow the instructions for using REXX EXECs. Check with your system programmer that the prototype invocation defines the prototype library.

Procedure: Starting the SDF II prototype dialog

Enter **sdf2p** on the command line to display the Select a Prototype Dialog panel:

```
                    SELECT A PROTOTYPE DIALOG

        1 DEFINE            Define prototype

        2 SIMULATE          Simulate the prototype

        3 PRINT             Get a printout of a prototype

        4 LIST              List the prototypes

        R REFERENCE         Obtain SDF II online reference

        X EXIT              Terminate SDF II dialog
```

This is what the options are used for:

1 DEFINE	Select this option to create a new prototype or to alter an existing prototype. Define the rules for a prototype here.
2 SIMULATE	Select this option to run a prototype.
3 PRINT	Select this option to print the rules that control the flow of panels in a prototype.
4 LIST	Select this option to list all or selected prototypes that exist, where they reside, and when they were last modified.
R REFERENCE	Select this option to access the online reference for the prototyping facility. The online reference covers all major topics within the prototype facility.
X EXIT	Select this option to leave the prototype facility.

Procedure: Identifying the prototype

Start this procedure from the Select a Prototype Dialog panel.

1. Enter **1** on the command line to go the Identify Prototype panel.

2. Enter the highlighted text to name the prototype you are about to create:

```
                              IDENTIFY PROTOTYPE

    NAME  . . . . . . . prot1

    LIBRARY . . . . . . DGIPROT

When creating a new prototype, identify the name of the skeleton prototype

    NAME  . . . . . . . _____

    LIBRARY . . . . . . _____
```

The first time you use this panel, the NAME field is blank. The next time you use this panel, the NAME field will contain the name you entered last.

The LIBRARY field must contain the DD name or FILEDEF name of the library where the prototype is to reside. The default for the library DD name is DGIPROT.

When you have defined a prototype with SDF II, you can base a new prototype on it. You enter the name and library of the existing prototype in the appropriate fields in the bottom half of the panel. Ignore the second NAME and LIBRARY fields for now.

The Specify Prototype panel

When you pressed the Enter key, SDF II displayed the Specify Prototype panel:

```
                        SPECIFY PROTOTYPE                          PROT1

PROTOTYPE . . . . . . . . . . . . . . . . . . . COLUMNS 1-4 OF 4, ROW 0 OF 0
    CURRENT  CONDITION ----------------- ACTION -------------------- NEXT PAN
 ''' TOP OF DATA ***************************************************************
 ''' END OF DATA ***************************************************************
```

The entries under each column are used as follows:

CURRENT   Define here the name of the current panel being displayed. When SDF II interprets a prototype, it searches the prototype definition for definitions with matching panel names. If the CURRENT column is blank, SDF II uses the previous nonblank entry in the CURRENT column.

CONDITION   Define here the condition to be checked. You can check whether the user has entered data into input fields or whether the user has pressed a program function key, or both.

ACTION   Specify here the actions that SDF II is to perform before it displays the next panel. It performs these actions if the condition specified under CONDITION is true. A blank CONDITION is always considered to be true.

NEXT PAN   Define what action is to be taken next. You can make the following entries:

- A name indicates the next panel to be displayed.

- The keyword *continue* allows you to continue adding further actions and conditions to the current rule of the current panel.

- The keyword *end* finishes the prototype.

- The name of an ISPF variable contains the identifier of the next panel to be displayed.

- No entry means that the next rule is processed.

When SDF II interprets a prototype, it searches the prototype definition line by line for a rule that fulfills both of the following:

- The name of the currently displayed panel is in the CURRENT column.

- The expression in the CONDITION column is true.

The first statement that meets these conditions is processed. Then the action in NEXT PAN is performed, if specified; otherwise, the next rule is processed.

You can use the following line commands in this dialog: **a** (after), **b** (before), **c** (copy), **cc** (copy block), **d** (delete), **dd** (delete block), **i** (insert), **m** (move), **mm** (move block), **r** (repeat), **rr** (repeat block), and **/** (make this the current line).

To customize the Specify Prototype panel, enter the **view** panel command. This takes you to another panel where you can specify which columns are displayed and their width.

The prototype rules

Each line in the Specify Prototype dialog constitutes one rule. You can specify a rule over several physical lines by entering **continue** in the NEXT PAN column.

Defining a prototype is based upon the following syntax for each rule:

- Enter the name of the first panel to be displayed, when the prototype definition is interpreted, in the CURRENT column of the first line of the prototype definition.

- To perform a certain action before displaying the first panel, leave the CURRENT and the CONDITION columns of the first line blank. The specified action is then performed, and the panel named in the NEXT PAN column is displayed.

- If you want SDF II to unconditionally perform a certain action after a panel has been displayed, leave the CONDITION column blank.

- If you want to unconditionally perform a certain action before checking further for conditions, leave CONDITION and NEXT PAN columns blank. Just enter the name of the current panel, and the action to be performed.

- To finish running the prototype when a certain condition is met, specify **end** in the NEXT PAN column.

- To specify the name of the next panel to be displayed, you can also enter the name of an ISPF variable that contains that panel name in the NEXT PAN column. Prefix the name of the ISPF variable with an ampersand and put the name of the variable in the ISPF shared pool.

- To test whether a function key has been pressed, use the special variable **dgipfkey**. For example, use **dgipfkey='pf03'** to check whether PF3 has been

pressed. To test whether the the Enter key has been pressed, use **dgipfkey=' '**, in a REXX EXEC, or **dgipfkey=''**, in a CLIST.

- To show sample values on the next panel, specify **dgisampl='on'** in the ACTION column. **dgisampl** is reset to **off** after the next panel has been displayed. If **dgisampl** is set to **off** the current value of a field is displayed. The current value is either the initial value of the field or the value that was in a field of the same name in a previous panel.

- To call your own routine, specify **call** *myprog*, where *myprog* is the name of your own routine.

Defining actions and conditions using REXX EXECs

This section is for users of REXX EXECs. CLIST users can go to the next section.

You can specify any valid REXX EXEC expression in the CONDITION column and any valid REXX EXEC statement in the ACTION column.

Conditions can be made up of several expressions that may be included in parentheses. To test for a certain value, place the value within quotes.

To perform more than one action, you must separate the actions by semicolons (for example, ZCMD=' '; MSG=' '). You can specify any valid REXX EXEC statement. To call your own REXX EXEC, specify **call** followed by the REXX EXEC name. To assign values to variables, place the values within quotes. Do not use a semicolon in any character string assigned to variables.

Values assigned to variables cannot be longer than 256 characters. Dynamic areas cannot be longer than 256 characters because they are represented by a variable.

REXX EXEC users should bypass the next section.

Defining actions and conditions using CLISTs

This section is for users of CLISTs. REXX EXEC users should go to the next section.

A *condition* is a simple CLIST expression. An *action* is either an assignment statement of the form *variable = expression*, or a call to a user-written routine of the form *call routine*. When you use CLISTs, you must follow certain rules when specifying *conditions* and when specifying *actions*.

The following restrictions apply when you specify *conditions*:

- You can join only two expressions with *and* or *or* (for example, ZCMD<'1' & ZCMD>'3').

- The only symbol allowed for *and* is &. The only symbol allowed for *or* is |.

- Do not include blanks in the expressions.

- To test a field for blanks, specify two consecutive single quotes (for example, ZCMD='').

- To test a string, enclose it in single quotes.

- Do not include blanks or any of these other characters in strings: =, <, >, or ¬.

- Do not use 'STR' as a literal as it is a reserved keyword.

The following restrictions apply when you specify *actions*:

- You can specify as many actions as you wish, separated by semicolons.

- If you call your own routine, you cannot pass positional or keyword parameters to it.

- Strings assigned to variables must not contain parentheses ().

- Do not assign values, such as the TSO SYSDATE variable, that can be interpreted as arithmetic expressions to a variable field. The TSO SYSDATE variable contains the (/) division symbol.

Procedure: Building the first simulative prototype

1. Enter the highlighted text to create 18 new lines in which to build the prototype:

```
i18 TOP OF DATA ***
''' END OF DATA ***
```

2. Enter the highlighted text to build the prototype:

```
                          SPECIFY PROTOTYPE                        PROT1

PROTOTYPE . . . . . . . . . . . . . . . . . . .  COLUMNS 1-4 OF 4, ROW 1 OF 18
      CURRENT  CONDITION ----------------- ACTION -------------------- NEXT PAN
'''  TOP OF DATA ***************************************************************
'''  prot00 _____  msg='' _____
'''  _____  cmd='1' _____  cmd='' _____  prot10 ____
'''  _____  cmd='2' _____  cmd='' _____  prot20 ____
'''  _____  cmd='3' _____  cmd='' _____  prot30 ____
'''  _____  dgipfkey='pf03' _____                            end _____
'''  _____                         msg='wrong input' _____  prot00 ____
'''  prot10 _____  msg='' _____
'''  _____  cmd='2' _____  cmd='' _____  prot20 ____
'''  _____  cmd='3' _____  cmd='' _____  prot30 ____
'''  _____  dgipfkey='pf03' _____                            prot00 ____
'''  prot20 _____  msg='' _____
'''  _____  cmd='1' _____  cmd='' _____  prot10 ____
'''  _____  cmd='3' _____  cmd='' _____  prot30 ____
'''  _____  dgipfkey='pf03' _____                            prot00 ____
'''  prot30 _____  msg='' _____
'''  _____  cmd='1' _____  cmd='' _____  prot10 ____
'''  _____  cmd='2' _____  cmd='' _____  prot20 ____
'''  _____  dgipfkey='pf03' _____                            prot00 ____
'''  END OF DATA ***************************************************************
```

Note: You can save your work at any stage by typing **save** on the command line and pressing the Enter key. SDF II stores your work and allows you to continue working in the current dialog.

For each panel name listed under the CURRENT column, similar conditions and actions apply. For example, for panel PROT00:

Line 1   Specifies that PROT00 is to be displayed and the message field cleared thereafter; then go on to the next rule.

Line 2   Specifies that a 1 entered in the **cmd** field of the panel causes SDF II to display panel PROT10 with a blank entry in the **cmd** field.

Line 3   Specifies that a **2** entered in the **cmd** field of the panel causes SDF II to display panel PROT20 with a blank entry in the **cmd** field.

Line 4   Specifies that a **3** entered in the **cmd** field of the panel causes SDF II to display panel PROT30 with a blank entry in the **cmd** field.

Line 5   Specifies that if the End key (PF3) is pressed, the prototype session is finished.

Line 6   Specifies that any other entry is not valid, and SDF II will display the message *wrong input* in the **msg** field.

Press the Return key (PF4) before continuing. This saves your work and SDF II displays the Select a Prototype Dialog panel.

Procedure: Running the first simulative prototype

Start this procedure from the Select a Prototype Dialog panel.

1. Enter **2** on the command line to display the Identify Prototype panel.

2. Enter **prot1** in the first NAME field to indicate to SDF II which simulative prototype you want to run.

3. SDF II displays the first panel of your simulative prototype. When you enter the appropriate option on the command line of this and the other panels of your prototype SDF II displays these panels of your prototype. Data that you enter into the fields is not stored. You can also use the End key (PF3) to display your first panel at any time.

4. After testing your first simulative prototype, return to the first panel of the prototype with the End key (PF3).

5. Press the Return key (PF4) to return to the Select a Prototype Dialog panel.

Procedure: Building the second simulative prototype

In this procedure, you build the second prototype which displays sample values. You use the first prototype as the *skeleton* for the second prototype. Sample values displayed in the fields of a panel help a prototype to look more like a final application. In the second simulative prototype, you define sample values in addition to the rules you defined for the first simulative prototype.

1. Enter **1** on the command line to display the Identify Prototype panel.

2. Enter the highlighted text to base your second simulative prototype on the previously created one.

```
                              IDENTIFY PROTOTYPE


    NAME  . . . . . . . . prot2

    LIBRARY . . . . . . . DGIPROT

    When creating a new prototype, identify the name of the skeleton prototype

    NAME  . . . . . . . . prot1

    LIBRARY . . . . . . . DGIPROT
```

3. Enter the highlighted text to build the second simulative prototype. (Use the I line command to insert lines where needed.)

```
                       SPECIFY PROTOTYPE                          PROT1

PROTOTYPE . . . . . . . . . . . . . . . . . . . . COLUMNS 1-4 OF 4, ROW 1 OF 21
     CURRENT  CONDITION ----------------- ACTION -------------------- NEXT PAN
'''  TOP OF DATA ***************************************************************
'''  PROT00 _____ MSG='' _____
'''         _____ CMD='1' _____ CMD='';_____ continue
'''         _____ dgisampl='on' _____ prot10
'''         _____ CMD='2' _____ CMD='';dgisampl='on' ___ PROT20
'''         _____ CMD='3' _____ CMD='';dgisampl='on' ___ PROT30
'''         _____ DGIPFKEY='pf03' _____ _____ END
'''         _____ MSG='WRONG INPUT' _____ PROT00
'''  PROT10 _____ MSG='' _____
'''         _____ CMD='2' _____ CMD='';dgisampl='on' ___ PROT20
'''         _____ CMD='3' _____ CMD='';dgisampl='on' ___ PROT30
'''         _____ DGIPFKEY='pf03' _____ _____ PROT00
'''  PROT20 _____ MSG='' _____
'''         _____ CMD='1' _____ CMD='';dgisampl='on' ___ PROT10
'''         _____ CMD='3' _____ CMD='';dgisampl='on' ___ PROT30
'''         _____ DGIPFKEY='pf03' _____ _____ PROT00
'''  PROT30 _____ MSG='' _____
'''         _____ CMD='1' _____ CMD='';dgisampl='on' ___ PROT10
'''         _____ CMD='2' _____ CMD='';dgisampl='on' ___ PROT20
'''         _____ cmd¬='' & _____ _____ continue
'''         _____ cmd>'2' _____ MSG='wrong command' ____ prot30
'''         _____ DGIPFKEY='pf03' _____ _____ PROT00
'''  END OF DATA ***************************************************************
```

The second prototype includes the statement **dgisampl = 'on'**. This statement instructs SDF II to display sample values in fields.

The **continue** keyword is used on lines 2 and 19. This is what the statement means:

Line 2   The action resulting from entering 1 in the CMD field (display a blank in the CMD field of panel PROT10), is continued on the next line (display sample values).

Line 19 The condition defined on this line (if the value in the **cmd** field is not blank), is continued on the next line (if the value in the **cmd** field is greater than **2**).

4. Press the Return key (PF4) to save and return to the Select a Prototype Dialog panel.

Procedure: Running the second simulative prototype

Start this procedure from the Select a Prototype Dialog panel.

1. Enter **2** on the command line to display the Identify Prototype panel.

2. Enter the highlighted text.

```
NAME . . . . . . . prot2
```

Enter the name of the simulative prototype you want to run.

3. SDF II now displays the first panel of your second simulative prototype. Enter the appropriate option on the command line of this and the other panels of your prototype. SDF II displays the panels of your prototype.

Sample data is displayed in the fields of the Order Entry, Order Update, and Order List panels. You can enter data in these fields. However, data entered in a simulative prototype panel is not stored. Therefore, any data you enter is lost when you enter a command to display another panel.

You can also use the End key (PF3) to display your first panel at any time.

After testing your first simulative prototype, return to the first panel of the prototype with the the End key (PF3).

4. Press the Return key (PF4) to return to the Select a Prototype Dialog panel.

Chapter 4. Operational prototype example

An operational prototype looks and works more like the application program is expected to look and work than do simulative prototypes. Values that you enter into fields on the panels are stored in a database.

The operational prototype described in this chapter uses an ISPF table as a database, but you could use any suitable alternative.

Adding logic to the prototype

An operational prototype also contains a limited amount of program logic. This logic controls the flow of panels in the prototype, depending on the entries made in the fields on the panels and on the information stored in the database.

Define your routines for the logic used by the operational prototype, by using either REXX EXECs or CLISTs. Then call your routine from the ACTION field, specifying:

CALL *myprog*,

where *myprog* is the name of your routine.

In your routine, you can access the panel's variables by means of the ISPF shared pool. Use the VGET service to retrieve the variables from the shared pool, and the VPUT service to return them to the shared pool.

If your routine initializes variables that are contained in the next panel, you must tell SDF II to retrieve these variables from the shared pool. To do this, store a list of these variables in the variable *DGINL1* in the shared pool as shown below:

For REXX EXECs:

```
DGINL1='(var1, var2,...)'
'ISPFEXEC VPUT DGINL1 SHARED'
```

For CLISTs:

```
SET &DGINL1=&STR((var1, var2,...))
ISPFEXEC VPUT DGINL1 SHARED
```

where *var1, var2,...* are the names of the variables.

An example of this processing can be seen in the supplied DGIWXRTR REXX and DGIWCRTR CLIST routines.

The routines for the operational prototype

There are three routines used in the operational prototype. These routines are supplied in two formats, CLIST and REXX EXEC, in the SDF II sample library.

The **REXX** routines are as follows:

DGIWXSTR   To store data

DGIWXRTR   To retrieve the stored data

DGIWXUPD   To update the data.

The **CLIST** routines are as follows:

DGIWCSTR   To store data

DGIWCRTR   To retrieve the stored data

DGIWCUPD   To update the data.

Check with your system programmer that these routines are accessible to you.

Procedure: Identifying the operational prototype

Start this step from the Select a Prototype Dialog panel.

1. Enter **1** on the command line to display the Identify Prototype panel.

2. Enter the following highlighted text:

```
                              IDENTIFY PROTOTYPE

    NAME  . . . . . . . prot3

    LIBRARY . . . . . . DGIPROT

When creating a new prototype, identify the name of the skeleton prototype

    NAME  . . . . . . . prot1

    LIBRARY . . . . . . DGIPROT
```

The name of your first simulative prototype is used as the *skeleton* of your operational prototype. When you press the Enter key, SDF II displays the Specify Prototype panel.

Procedure: Building the operational prototype

In addition to the rules of the first simulative prototype, the operational prototype includes statements that call the appropriate routines that provide the programming logic for the operational prototype.

1. Make the changes shown in highlighted form. Use the I command to insert lines where needed.

For REXX users:

	CURRENT	CONDITION ----------------	ACTION --------------------	NEXT PAN
'''	PROT00		MSG=''	
'''		CMD='1'	CMD=''	PROT10
'''		CMD='2'	CMD=''	PROT20
'''		CMD='3'	call dgiwxrtr	PROT30
'''		DGIPFKEY='pf03'		END
'''			MSG='WRONG INPUT'	PROT00
'''	PROT10		MSG=''	
'''		CMD='2'	CMD=''	PROT20
'''		CMD='3'	call dgiwxrtr	PROT30
'''		DGIPFKEY='pf03'		PROT00
'''		custno¬=''	call dgiwxstr	prot10
'''	PROT20		MSG=''	
'''		cmd='1'	CMD=''	PROT10
'''		CMD='3'	call dgiwxrtr	PROT30
'''		DGIPFKEY='pf03'		PROT00
'''		custno¬=''	call dgiwxupd	prot20
'''	PROT30		MSG=''	
'''		CMD='1'	CMD=''	PROT10
'''		CMD='2'	CMD=''	PROT20
'''		DGIPFKEY='pf03'		PROT00
'''		dgipfkey='pf07'	call dgiwxrtr	prot30
'''		dgipfkey='pf08'	call dgiwxrtr	prot30

For CLIST users:

	CURRENT	CONDITION ----------------	ACTION --------------------	NEXT PAN
'''	PROT00		MSG=''	
'''		CMD='1'	CMD=''	PROT10
'''		CMD='2'	CMD=''	PROT20
'''		CMD='3'	call dgiwcrtr	PROT30
'''		DGIPFKEY='pf03'		END
'''			MSG='WRONG INPUT'	PROT00
'''	PROT10		MSG=''	
'''		CMD='2'	CMD=''	PROT20
'''		CMD='3'	call dgiwcrtr	PROT30
'''		DGIPFKEY='pf03'		PROT00
'''		custno¬=''	call dgiwcstr	prot10
'''	PROT20		MSG=''	
'''		cmd='1'	CMD=''	PROT10
'''		CMD='3'	call dgiwcrtr	PROT30
'''		DGIPFKEY='pf03'		PROT00
'''		custno¬=''	call dgiwcupd	prot20
'''	PROT30		MSG=''	
'''		CMD='1'	CMD=''	PROT10
'''		CMD='2'	CMD=''	PROT20
'''		DGIPFKEY='pf03'		PROT00
'''		dgipfkey='pf07'	call dgiwcrtr	prot30
'''		dgipfkey='pf08'	call dgiwcrtr	prot30

2. Press the Return key (PF4), to save the prototype and return to the Select a Prototype Dialog panel.

Procedure: Run the operational prototype

Start this procedure from the Select a Prototype Dialog panel.

1. Type **2** on the command line to display the Identify Prototype panel.

2. Enter the name of the simulative prototype you want to run as shown below:

   NAME . . . . <u>**prot3**</u>

3. SDF II now displays the first panel of your operational prototype. Enter the appropriate option on the command line of this and the other panels of your prototype. SDF II displays the panels of your prototype.

   You can now enter data in the fields of the Order Entry panel. The data is stored in the database. You can also change the details of orders on the Order Update panel, and list orders on the Order List panel, scrolling blocks of four orders Forward (PF8) and Backward (PF7).

   You can also use the End key (PF3) to display your first panel at any time.

   After testing your first operational prototype, return to the first panel of the prototype with the End key (PF3).

4. Press the Return key (PF4) to return to the Select a Prototype Dialog panel.

You can leave the prototype facility of SDF II from this panel by entering **x** on the command line.

Chapter 5. Printing and library listing facilities

SDF II provides facilities for the following:

- Printing the prototype contents

- Listing the library contents in various ways.

These facilities are covered in this chapter.

Procedure: Printing the contents of your prototype

Start this procedure from the Select a Prototype Dialog panel.

1. Enter **3** on the command line to go to the Specify Print Utility Parameters panel.

2. Enter the highlighted text to specify which prototype is to be printed. SDF II automatically sends the contents of your prototype to the system printer. When the print utility finishes its task, it displays the message Printed in the top right-hand corner of the panel.

```
                    SPECIFY PRINT UTILITY PARAMETERS

Identification of prototype to be printed
    NAME  . . . . . . prot1

    LIBRARY . . . . . DGIPROT

Format of listing
    FORMAT  . . . . . 1          1 - Standard printer
                                 2 - Input to DCF
                                 3 - DBCS Printer

Destination for format 2 only
    DCF DS NAME . . . _____
```

Note: The only difference for VM users is that the Destination for format 2 only section at the bottom of the displayed panel is not present.

3. Press the End key (PF3) to return to the Select a Prototype Dialog panel.

Procedure: Listing the contents of your libraries

Start this procedure from the Select a Prototype Dialog panel.

1. Enter **4** on the command line to display the Specify Search Argument panel.

2. Enter **pfshow off** on the command line to turn off the PF key fields display at the bottom of the panel. The line commands that are available on the List Prototypes panel will then appear when this panel is displayed following the next step.

3. Enter the highlighted text to request a list of all prototypes, in prototype name order, that begin with the letter **p** and are stored in library **dgiprot**.

23

```
                         SPECIFY SEARCH ARGUMENT

       NAME  . . . . . . . . p*

       LIBRARY . . . . . . . dgiprot

       SORT SEQUENCE . . . . n_            N - Prototype name
                                           L - Library
                                           M - Modification date
```

The library details are returned in the following way:

```
                             LIST PROTOTYPES

       SEARCH ARGUMENT: NAME = P*       LIBRARY = DGIPROT   SORT SEQUENCE = N

       PROTOTYPE LIST  . . . . . . . . . . . . . . . . . COLUMNS 1-5 OF 5, ROW 1 OF 3
             NAME --- LIB ---- OPERANDS -------- LAST MODIFIED --- USER ---
       '''  PROT1    DGIPROT _____ 92/01/18 15:39:40 USERID1
       '''  PROT2    DGIPROT _____ 92/01/17 09:40:40 USERID1
       '''  PROT3    DGIPROT _____ 92/01/08 13:41:30 USERID1




       '''  _____ _____ _____ <=== WORK HERE ON ANOTHER OBJECT

       Line command: C = Copy  D = Delete  R = Rename  P = Print
                     E = Edit  SIM = Simulate  = = Repeat command
```

The free input line near the bottom of the panel can be used to identify any prototype and action that you wish to perform on it.

The line commands can be used to request the selected action be performed on a prototype without the need to first return to the Select a Prototype Dialog panel. The OPERANDS field is provided for the entering of any necessary operands for a line command.

4. Press the Return key (PF4) to return to the Select a Prototype Dialog panel.

Enter x on the command line to leave SDF II.

Glossary of terms and abbreviations

Glossary terms are defined as they are used in this book. Some definitions have been taken from *American National Standard Dictionary for Information Systems*, in which case they are marked with (A); other definitions are from the *Information Technology Vocabulary*, in which case they are marked with an (I). Definitions without source labels are IBM definitions. If you cannot find the term you are looking for, refer to the index, the online reference index, or to the *IBM Dictionary of Computing*, SC20-1699.

A

AD. Application development.

application. A collection of software components used to perform specific types of user-oriented work on a computer. Typical examples are payroll applications, airline seat-reservation systems, and stock-control systems.

application development (AD). The defining, writing, and testing of a program for a specific solution or application problem.

application prototype. A simulation in a predefined order of some or all panels used in an application. See operational prototype, prototype, and simulative prototype.

C

CLIST. Command list in MVS.

CMS. Conversational monitor system.

conversational monitor system (CMS). A virtual machine operating system that provides general interactive time sharing, problem solving, and program development capabilities, and operates only under control of the VM/370 VM control program.

D

data structure. In SDF II, a structure that is part of a panel. For output, it describes how data is provided by the application. For input, it describes how data is presented to the application.

DBCS. Double-byte character set.

DCF. Document composition facility.

DLIB. Distribution library.

double-byte character set (DBCS). A set of characters in which each character is represented by 2 bytes.

Languages such as Japanese, Chinese, and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets. Because each character requires 2 bytes, the typing, display, and printing of DBCS characters requires hardware and programs that support DBCS.

E

EXEC. An executable procedure that contains CMS or TSO commands and execution control statements, such as branches.

F

format. In SDF II, a part of a panel that defines how data appears on a screen. For output, it defines how data is presented on a screen. For input, it defines how data is entered in a screen by a user. A format can consist of different definitions for different device types. These definitions are called format instances.

G

generation. In SDF II, a process by which objects are created for use in the target systems or for prototyping the application.

I

Interactive System Productivity Facility (ISPF). An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, it provides a means of generating standard screen panels and interactive dialogs between the application programmer and terminal user.

ISPF. Interactive System Productivity Facility.

ISPF/PDF. Interactive System Productivity Facility/Program Development Facility.

L

library identifier. In SDF II, a single character that identifies a library that contains SDF II objects.

line command. In SDF II, a command used to request a function. It applies to a specific line or lines. Examples are C for Copy or M for Move.

line command area. An area on the left-hand side of a window where line commands are entered.

M

Multiple Virtual Storage (MVS). See MVS.

MVS. Multiple virtual storage. Implies MVS/370, the MVS/XA product, and the MVS/ESA product.

O

object. In SDF II, an item upon which actions are performed, such as a panel, panel group, partition set, AID table, or operator control table. It is stored in an SDF II library. Synonymous with specification object.

operational prototype. A model of an application program used to test or review simple functions, such as simple database access, scrolling, error reporting, and online help panels. See prototype and simulative prototype.

P

panel. (1) The information displayed at the same time on the screen of a display device. (2) In SDF II, an object that consists of formats, data structures, and various tables. Each panel has at least one format.

panel command. A command that affects a part of the panel, the whole panel, or the flow of SDF II. Panel commands are entered on the command line. They can be assigned to program function keys.

panel group. In SDF II, an object that contains a list of panel names and describes the properties of these panels.

PDS. Partitioned data set.

prototype. In SDF II, a model or preliminary implementation used to evaluate system design, or better understand or determine requirements. See application prototype, operational prototype, and simulative prototype.

R

Restructured Extended Executor (REXX). An interpretive language used to write command lists.

REXX. Restructured Extended Executor language.

S

SDF II. Screen Definition Facility II.

simulative prototype. An application model that uses a series of panels to test or review the primary flow of interactions between the application program and its users. The panels may display initial values and may accept data entered by a user. See operational prototype and prototype.

skeleton. An object used as a model when creating a new object.

T

target system. In SDF II, a system under which the application using an SDF II generated object can be run. For example, CICS/BMS or IMS/MFS.

U

user exit routine. A user-written routine that receives control at predefined user exit points. In SDF II, for example, it is a CLIST or an EXEC.

V

variable field. A field in which data may be changed by the application program or by the user. Contrast with constant field.

VM. Virtual machine. Implies VM/SP, VM/HPO, VM/XA, or VM/ESA.

SDF II publications

The SDF II Release 3 publications are:

Licensed Program Specifications, GH19-6115
Contains the product specifications and warranty information.
Audience: Data processing manager, system programmer.

General Information, GH19-6114
Summarizes the functions, uses, requirements, and advantages of SDF II.
Audience: Data processing manager, system programmer.

Installation and Migration Guide for MVS, SH19-6116
Describes how to install SDF II on an MVS system. It also explains how to migrate objects from the target systems and SDF/CICS into SDF II.
Audience: System programmer, application programmer.

Installation and Migration Guide for VM, SH19-6117
Describes how to install SDF II on a VM system. It also explains how to migrate objects from the target systems and SDF/CICS into SDF II.
Audience: System programmer, application programmer.

General Introduction, SH19-8165
Introduces SDF II to new users and explains how to define simple panels. It also explains the more advanced functions of SDF II to experienced users.
Audience: System programmer, application programmer, end user.

Primer for CICS/BMS Programs, SH19-6118
Explains how to use SDF II to develop objects for applications that run under CICS/BMS.
Audience: System programmer, application programmer, end user.

Primer for IMS/MFS Programs, SH19-6453
Explains how to use SDF II to develop objects for applications that run under IMS/MFS.
Audience: System programmer, application programmer, end user.

Primer for ISPF Programs, SH19-6119
Explains how to use SDF II to develop objects for applications that run under ISPF.
Audience: System programmer, application programmer, end user.

Primer for GDDM-IMD and CSP/AD Programs, SH19-6459
Explains how to use SDF II to develop objects for applications that run under GDDM-IMD or CSP/AD.
Audience: System programmer, application programmer, end user.

Preparing a Prototype, SH19-6458
Explains how to simulate an application using the prototype function of SDF II.
Audience: System programmer, application programmer.

Reference Summary, SX11-6088
Lists and explains SDF II line and panel commands. It also lists the main dialogs and functions of SDF II.
Audience: System programmer, application programmer, end user.

Diagnosis Guide, LY19-6251
Explains how to identify and report faults in SDF II to IBM support personnel.
Audience: System programmer.

Index

A
actions and conditions
  CLISTs   14
  REXX EXECs   14

B
books of SDF II   27

C
CLISTs   1, 14
command
  to start to prototype (sdf2p)   1, 11
communicating
  with SDF II   vii
continue keyword   13, 17

D
database
  function   1, 19
  programming logic   1
dgisampl statement   17
dialogs
  Identify Prototype   12
  List Prototypes   23
  Select a Prototype   11
  Specify Print Utility Parameters   23
  Specify Prototype   1, 12

E
end keyword   13
error messages
  message help   viii
EXECs   1, 14

F
field
  help   viii

H
help information   viii
help key   viii

I
IBM notices   v
IBM trademarks service marks   v
Identify Prototype
  field descriptions   12
  panel layout   12

I
initial values   1
ISPF
  services
    VGET   19
    VPUT   19
  table   19

J
join expressions
  | (or)   14
  & (and)   14

K
keyword
  continue   13
  end   13

L
library
  listing contents of   23
line command
  after (a)   13
  before (b)   13
  copy block (cc)   13
  copy (c)   13
  definition and usage   vii
  delete block (dd)   13
  delete (d)   13
  insert (i)   13
  move block (mm)   13
  move (m)   13
  repeat block (rr)   13
  repeat (r)   13
  / (set current line)   13
List Prototypes
  panel layout   24
long messages   viii

M
message help   viii
messages
  SDF II   vii
message, error   viii

O
online reference
  topics in   viii
operational prototype
  program logic   19

Communicating Your Comments to IBM

Screen Definition Facility II
Preparing a Prototype
Release 3

Publication No. SH19-6458-02

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:

  Your International Access Code + 43 + 1 + 21145-4490

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.

Readers' Comments

Screen Definition Facility II
Preparing a Prototype
Release 3

Publication No. SH19-6458-02

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply.

Name _____   Address _____

Company or Organization _____

Phone No. _____

IBM

```
PLACE
POSTAGE
STAMP
HERE
```

IBM Vienna Software Development Laboratory
Information Development
Department 00/705
Obere Donaustraße 95
A-1020 Vienna
Austria

IBM

Readers' Comments

Screen Definition Facility II
Preparing a Prototype
Release 3

Publication No. SH19-6458-02

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply.

Name _____   Address _____

Company or Organization _____

Phone No. _____

IBM

PLACE
POSTAGE
STAMP
HERE

IBM Vienna Software Development Laboratory
Information Development
Department 00/705
Obere Donaustraße 95
A-1020 Vienna
Austria

SH19-6458-02

Readers' Comments

Screen Definition Facility II
Preparing a Prototype
Release 3

Publication No. SH19-6458-02

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply.

Name _____   Address _____

Company or Organization _____   _____

Phone No. _____   _____

IBM

PLACE
POSTAGE
STAMP
HERE

IBM Vienna Software Development Laboratory
Information Development
Department 00/705
Obere Donaustraße 95
A-1020 Vienna
Austria

Readers' Comments

IBM

SH19-6458-02

IBM

Program Number: 5664-307 (VM)
5665-366 (MVS)

SH19-6458-02