Screen Definition Facility II for VSE

Administrator’s Guide

Release 6

SH12-6311-00

Screen Definition Facility II for VSE

Administrator's Guide

Release 6

SH12-6311-00

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page ix.

First Edition (December 1997)

This edition applies to Release 6 Modification Level 0 of Screen Definition Facility II for VSE, Program Number 5746-XXT, and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for readers’ comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Vienna Software
SDF Development/Service
Dept 00/174
Lassallestrasse 1
A–1020 Vienna
Austria
Europe

E-mail: kpesen@vnet.ibm.com

If you prefer to send comments by fax, use this number:

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

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

Contents

Figures

Notices

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Deutschland Informationssysteme GmbH, Department 3982, Pascalstrasse 100, 70569 Stuttgart, Germany. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Trademarks and service marks

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

BookManager
CICS
CICS/ESA
CICS/VSE
Common User Access
CUA
GDDM
IBM
IMS
MVS/ESA
MVS/XA
VisualGen

Other company, product, and service names, which may be denoted by a double asterisk (**) in this publication, may be trademarks of others.

About this book

This book is written for system administrators who will set up and maintain Screen Definition Facility II for VSE (SDF II) in a software development environment.

What you will find inside

This book provides the following information:

Chapter 1 Briefly outlines the objectives and principles of SDF II. It also explains the types of objects that SDF II uses and briefly discusses storage strategies.

Chapter 2 Explains how to customize the SDF II environment, including how to set up libraries for SDF II objects and generated output, and how to change the uppercase/lowercase translation table and the displayable screen characters translation table.

Chapter 3 Explains how to start SDF II in full-screen mode and in batch mode.

Chapter 4 Explains how to define installation defaults, device types, emphasis classes, and Common User Access (CUA) elements. It also explains how to adapt print utility skeletons, how to change the SDF2/DM keylists, and how to make your changes available to other users.

Chapter 5 Describes how to use the invocation interface, predefined panel elements, and generated output, and how to write user exit routines.

Chapter 6 Describes how to import object definitions into SDF II from the various target systems.

Appendix A Lists the device types supported by SDF II.

Appendix B Describes service procedures.

Appendix C Describes what you need to do when you encounter an error that you wish to communicate to IBM.

Glossary of terms and abbreviations
Contains a list of terms and their definitions used in SDF II.

SDF II publications
Contains a list of related publications.

At the end of the book is an index.

Highlighting conventions used in this book

Font	Convention
Bold	Names of panels, fields, and function keys; phrases or paragraphs containing important information; text or commands you are to type.
`Monospace`	Text displayed in the sample programs.
Italics	Words that are used for emphasis; variables in command strings.

Syntax conventions used in this book

Syntax shown in this book uses a simplified Backus Naur form. Syntax diagrams that might otherwise be over-complex and hard to understand quickly are separated into their major constituent parts.

[ ] Square brackets identify an optional syntactic unit, which you may use.

| The logical OR sign indicates that you can choose the syntactic unit to its left or right.

... An ellipsis indicates that the immediately preceding syntactic unit is repeated.

COMMAND Text shown in roman boldface font is a command or keyword.

variable Text shown in italic font represents a variable.

Chapter 1. Introduction

Screen Definition Facility II for VSE (SDF II) is an interactive application development tool that helps you develop and maintain screen objects on any IBM 3270 family or compatible display device. These SDF II screen objects are panels, panel groups, partition sets, AID tables, and control tables. They can be generated for use in application programs for various target systems and programming languages.

Using SDF II, you can define screen objects for following target systems:

CICS/BMS
IMS/MFS
ISPF
CSP/AD
VisualGen
GDDM–IMD

The programming languages in which you can generate data structures that describe screen objects are:

Assembler
C, C++
COBOL
PL/I
RPG

Note: SDF II VSE fully supports CICS/BMS, ISPF, and CSP/AD external source format. IMS/MFS, GDDM–IMD, and CSP/AD export data set objects can only be defined and prototyped using SDF II VSE Release 6; they cannot be imported into, or generated in, SDF II VSE Release 6. IMS/MFS and GDDM–IMD source objects defined using SDF II VSE Release 6 can be imported into, and generated in, SDF II MVS Release 4 or later, program number 5665-366, unless otherwise stated.

Types of objects

Objects used with SDF II are:

- SDF II objects
- Generated data structures
- Generated control block source
- Imported data
- Prototypes
- Print output

SDF II objects

These are objects produced by an SDF II editor. They are internal to SDF II and should be modified only by SDF II. SDF II objects are panels, panel groups, partition sets, AID tables, and control tables.

Figure 1 lists the types of SDF II objects and shows which object can be produced for which target system.

Figure 1. SDF II object types

Object	Type	IMS/MFS	CICS/BMS	GDDM–IMD	CSP/AD or VisualGen	ISPF
Panel	DGIPNL	√	√	√	√	√
Panel group	DGIGRP		√	√	√	
Partition set	DGIPST	√	√			
AID table	DGITBL	√		√		
Control table	DGIOCT	√				

Note: SDF II VSE cannot generate or import:

- SDF II objects defined for IMS/MFS or GDDM–IMD. To generate such objects, first move them to SDF II MVS Release 4 or later, unless otherwise stated.
- Export data sets defined for CSP/AD. Use, instead, external source format.

SDF II VSE Release 6 objects are compatible with SDF II MVS Release 4 objects.

Generated data structures

These are data structures produced by the generation function of SDF II to be included in application programs that run under the CICS/BMS or ISPF target system.

Generated control block source

These are objects produced by the generation function of SDF II for use in one of the SDF II target systems.

Generated control block source consists of CICS/BMS macros, CSP/AD external source format, or ISPF panels.

Figure 2 shows which control block source can be generated from which SDF II object and for which target system.

Figure 2. Generation of control block source

SDF II Object	CICS/BMS	CSP/AD or VisualGen	ISPF
Panel	Map	External source format	Panel
Panel group	Map set	External source format	
Partition set	Partition set		

Import data

This is control block source or SDF/CICS dump data created outside SDF II to be imported into SDF II. Import data can be CICS/BMS macros, CSP/AD external source format, extended external source format, ISPF panels, or SDF/CICS dump data.

Prototypes

These are objects, produced by the SDF II prototype facility, that are to be used in SDF II to simulate the flow of control and the function of application programs. They consist of SDF2/DM tables and panels, and REXX EXECs.

Print output

These contain output from the print utilities and messages from utilities, such as the generation or import utility. See also “Destination of print output” on page 17.

Where objects are stored

All SDF II objects and prototypes are stored in VSE/ESA sublibraries.

You may decide to group your objects in a particular way, such as by project, by target system, or by development level. Define one or more object libraries, to meet the requirements of your projects. Each SDF II user can work with up to nine libraries at the same time. The standard VSE access control applies for all SDF II libraries.

SDF II dialogs

To help you more easily create and manage objects, SDF II provides interactive dialogs. These dialogs are controlled by the dialog management component of SDF II, SDF2/DM, which can also be used to build and run prototypes. SDF2/DM provides services for displaying panels, for file tailoring, and for processing variables, tables, and messages. (For more information about SDF2/DM and its services, refer to *SDF II Run-Time Services*.)

The SDF II dialogs consist of those used to customize SDF II and those used to develop and maintain SDF II objects.

SDF II customization dialogs

The following SDF II dialogs are intended for the SDF II administrator:

Device type editor
A device type defines the characteristics of devices referred to in the panel, panel group, and partition set editor dialogs. You can group devices in device lists.

Emphasis class table definition
You can group attributes into emphasis classes. In the panel editor, users can assign emphasis classes instead of assigning single attributes.

CUA attribute definition
You can change the attributes of CUA panel element types.

National language translation: SDF II messages, tutorial topics, help panels, function panels, and print utility skeletons, which are contained in SDF2/DM message and panel libraries, can be translated using any standard text editor.

SDF II object development dialogs

The following SDF II dialogs are available to all SDF II users:

Panel editor
The main components of a panel are its format and its data structure. The format describes how data is to be presented on a particular device. The data structure describes how data is communicated from and to the application program. Other components of a panel are the panel characteristics and tables containing symbols with special meanings for the definition process. SDF II distinguishes between general panel characteristics and those specific to a target system.

Panel group editor

A panel group is an object containing information required by CICS/BMS, GDDM–IMD, CSP/AD, or VisualGen run-time services. The components of a panel group are its characteristics, a list of all panels to be contained, and, for GDDM–IMD, a list of all programmed symbol sets used explicitly or implicitly by the panel group.

Partition set editor

A partition set defines how the screen of an IBM 8775, 3290, or 3180 display device is split into a number of separately controlled areas. The components of a partition set are its characteristics and a list of all partitions contained in the partition set.

AID (attention identifier) table editor

An AID table maps operator actions to values, which are then returned to the application program.

Control table editor

A control table defines a sequence of conditional operations and their associated control or branching functions.

Generation

Panels, panel groups, and partition sets can be generated for use in the target systems as follows:

Target system	Generated objects
CICS/BMS	CICS/BMS macros, application data structures
ISPF	ISPF panels, data structures
CSP/AD	CSP/AD external source format
VisualGen	CSP/AD external source format

User exits are provided to invoke an EXEC after generation, to further process the generated output.

List objects

The List Objects panel gives the names and types of all objects that satisfy the generic search argument. You can perform actions on these objects, such as invoking the editors, generating objects, or printing.

Specify libraries

In the Specify Libraries dialog, you specify the libraries to be accessed and the order in which they are to be searched. These libraries are then available for all SDF II dialogs. You access them through the SDF II dialog manager (SDF2/DM).

Utilities

SDF II includes utilities to help you manage SDF II objects:

- Use the *print utility* to print a representation of any SDF II object. You can select whether the output is for the system printer (SYSLST) or for a DBCS printer.
- Use the *import utility* to import the following into an SDF II-supported library:
  - Maps, map sets, and partition sets defined with CICS/BMS macros
  - Panels defined in the panel syntax of ISPF Version 2 or later
  - Maps and map groups defined in CSP/AD external source format or extended external source format
  - Maps, map sets, and partition sets defined and dumped using SDF/CICS

- Use the *conversion utility* to convert SDF II objects defined for one target system to a format suitable for the use in another target system.
- Use the *print online reference utility* to print all or part of the online reference.
- Use the *panel construction utility* to construct a panel from a list of predefined panel elements.

Profile

You can define defaults that are used in SDF II dialogs to tailor the environment to your specific needs with the profile editor.

Extraction Utility

Use the extraction utility to extract information about a panel and its fields, such as the field's position, its length, or its contents. This information is passed to a user exit where any further processing can be done.

Modification Utility

The modification utility is similar to the extraction utility but offers an extra function in the user exit with which you can modify the contents of fields, copy formats, and change the device type of a panel.

SDF II application prototype

Use the SDF II application prototype function to review the flow of panels, before anyone writes a line of code. By involving eventual users of the application in such reviews, you can get early feedback, which can then be incorporated into the design of the application without anyone having to rewrite the application program code.

There are two main types of application prototypes:

- A *simulative prototype*, which is used early in the development cycle. It consists of a series of panels, representing a first approximation of the application's panels.
- An *operational prototype*, which is a limited function version of the application program. It can be as simple or complex as necessary.

SDF II supports both simulative and operational prototypes. It provides all the functions needed to define and run a prototype. By adding your own routines, you can make the prototype as complex as you want, thereby moving towards an operational prototype.

Prototyping is discussed in *SDF II General Introduction*. How to fully code a prototype is discussed in *SDF II Run-Time Services*.

Online reference

The online reference contains all the information you need for working with SDF II. How to use it, and how to obtain a printed copy, is discussed in *SDF II General Introduction*.

Chapter 2. Adapting the SDF II installation

This chapter explains how to customize SDF II to the needs of your installation. It also explains the types and storage locations of objects.

An interactive SDF II session requires a CICS/VSE transaction and an associated VSE/ESA batch partition. A batch SDF II job runs in a batch partition.

Figure 3 shows the SDF II execution environment. The shaded areas indicate those parts of the environment that you can customize.

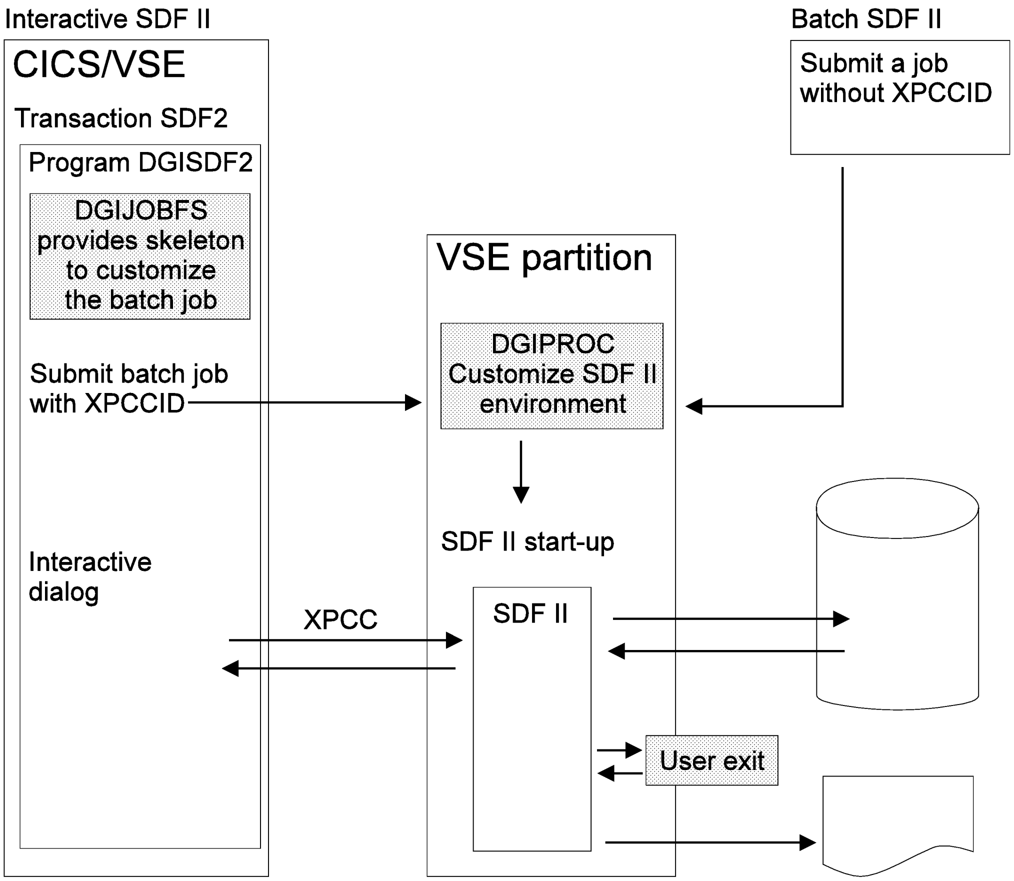

Interactive SDF II
CICS/VSE
Transaction SDF2
Program DGISDF2
DGIJOBFS provides skeleton to customize the batch job
Submit batch job with XPCCID
Interactive dialog
XPCC
Batch SDF II
Submit a job without XPCCID
VSE partition
DGIPROC Customize SDF II environment
SDF II start-up
SDF II
User exit

Figure 3. The SDF II execution environment. Customizable parts of the environment are highlighted.

Customizing the full-screen environment

Full-screen SDF II is started as a CICS/VSE transaction. To enable users to invoke full-screen SDF II, perform the following customization:

1. Make sure your SDF II installation library (default PRD2.PROD) is defined in the CICS/VSE start-up job library search chain.

   Note: The SDF II object libraries are not required to be in the CICS/VSE LIBDEF chain.

2. Define the SDF II transaction to CICS/VSE by either of these methods:
   - Online—see “Defining the SDF II transaction online”
   - Batch—see “Defining the SDF II transaction in batch mode”

Defining the SDF II transaction online

Use resource definition online (CEDA) to add the following definitions:

1. Define the programs:

   ```
   CEDA DEF PROG(DGISDF2)  G(SDF2) LAN(ASSEMBLER)
   CEDA DEF PROG(DGIJOBFS) G(SDF2) LAN(ASSEMBLER)
   ```

2. Define the profile:

   ```
   CEDA DEF PROF(DGIPROF)  G(SDF2) SCR(ALTERNATE)
   ```

3. Define the transaction:

   ```
   CEDA DEF TR(SDF2) G(SDF2) PROG(DGISDF2) PROF(DGIPROF)
   ```

4. Install the SDF II group:

   ```
   CEDA I G(SDF2)
   ```

5. Activate the SDF2 group whenever CICS/VSE is started:

   ```
   CEDA ADD GROUP(SDF2) LIST(VSELIST)
   ```

   If a value other than VSELIST was specified for the GRPLIST parameter during CICS/VSE system initialization, specify that value, instead of VSELIST, for the LIST parameter.

Defining the SDF II transaction in batch mode

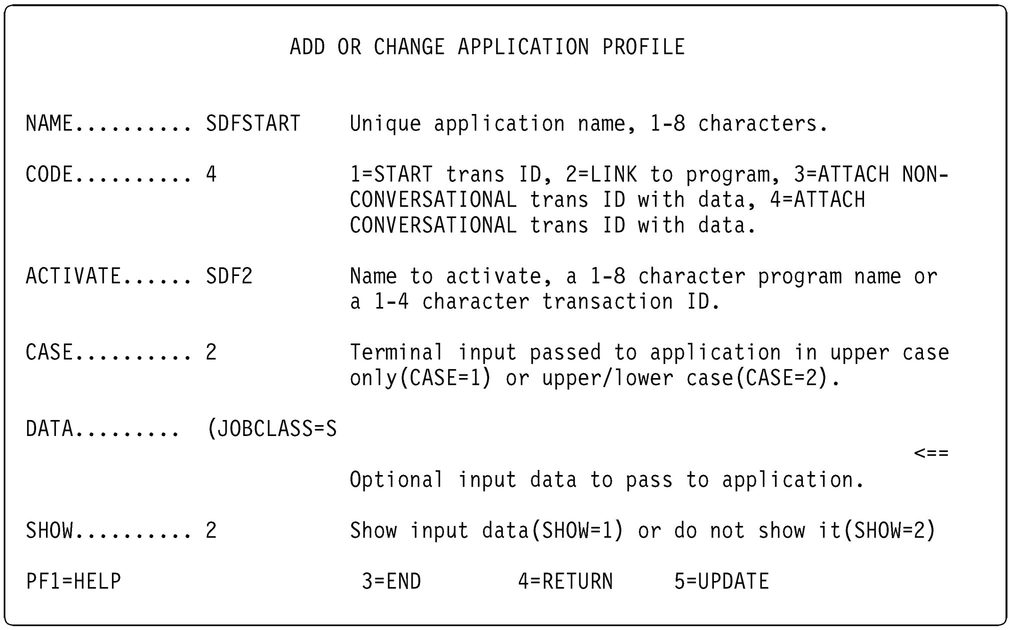
Use batch resource definition (DFHCSDUP) to add the following definitions:

```
* $$ JOB JNM=DFHCSDUP
// JOB DFHCSDUP
// EXEC DFHCSDUP
 DEFINE PROFILE(DGIPROF)  GROUP(SDF2) SCRNSIZE(ALTERNATE)
 DEFINE PROGRAM(DGIJOBFS) GROUP(SDF2) LANGUAGE(ASSEMBLER)
 DEFINE PROGRAM(DGISDF2)  GROUP(SDF2) LANGUAGE(ASSEMBLER)
 DEFINE TRANSACTION(SDF2) GROUP(SDF2) PROGRAM(DGISDF2) PROFILE(DGIPROF)
 ADD    GROUP(SDF2)       LIST(VSELIST)
/*
/&
* $$ EOJ
```

If a value other than VSELIST was specified for the GRPLIST parameter during CICS/VSE system initialization, specify that value, instead of VSELIST, for the LIST parameter.

Adding Interactive Interface definitions (optional)

Optionally define SDF II to the Interactive Interface. Define an application profile, then add it to an Interactive Interface panel.

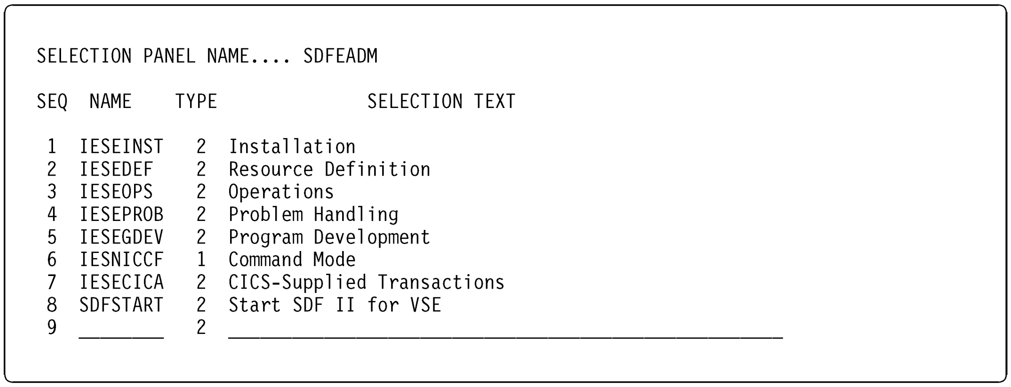


```
                      ADD OR CHANGE APPLICATION PROFILE


NAME.......... SDFSTART    Unique application name, 1-8 characters.

CODE.......... 4           1=START trans ID, 2=LINK to program, 3=ATTACH NON-
                           CONVERSATIONAL trans ID with data, 4=ATTACH
                           CONVERSATIONAL trans ID with data.

ACTIVATE...... SDF2        Name to activate, a 1-8 character program name or
                           a 1-4 character transaction ID.

CASE.......... 2           Terminal input passed to application in upper case
                           only(CASE=1) or upper/lower case(CASE=2).

DATA.........  (JOBCLASS=S
                                                                          <==
                           Optional input data to pass to application.

SHOW.......... 2           Show input data(SHOW=1) or do not show it(SHOW=2)

PF1=HELP                    3=END        4=RETURN     5=UPDATE
```

In the DATA field, you can specify any of the SDF II invocation groups that are described under “Starting SDF II in full-screen mode” on page 20.

Example: Add the selection to an Interactive Interface panel

In this example, panel IESEADM is used as the source to add the SDF II selection.



```
SELECTION PANEL NAME.... SDFEADM

SEQ  NAME    TYPE             SELECTION TEXT

 1  IESEINST   2  Installation
 2  IESEDEF    2  Resource Definition
 3  IESEOPS    2  Operations
 4  IESEPROB   2  Problem Handling
 5  IESEGDEV   2  Program Development
 6  IESNICCF   1  Command Mode
 7  IESECICA   2  CICS-Supplied Transactions
 8  SDFSTART   2  Start SDF II for VSE
 9  ________   2  ____________________________________________________
```

For more information about Interactive Interface tailoring, refer to *VSE/ESA Administration* and to *VSE/ESA Planning*.

Defining libraries

The SDF II objects are stored in VSE/ESA sublibraries that can be accessed by all SDF II users. DLBL and EXTENT information for the libraries must be available for interactive and batch SDF II jobs. Any DLBL or EXTENT JCL statements not available in the standard labels must be added to the SDF II start-up procedure DGIPROC.PROC.

This section explains how to define the work library, the object libraries, and the autosave library.

Defining the SDF II work library

SDF II requires a sublibrary with the name DGIWORK.DGI to store temporary data. The number of library blocks required depends on the number of concurrent users and the size of SDF II objects being processed.

The print utility also uses DGIWORK.DGI. To prepare the SDF II online reference, the print utility requires about 1200 library blocks. The sample uses 60 tracks on an IBM 3380 device, which is equivalent to 1860 library blocks of 1 KB. (K equals 1000.) Use this value to obtain an initial estimate of the size of the work library.

Sample job to define the library

Modify the following job control statements to meet the requirements of your installation.

```
* $$ JOB JNM=DEFLIBR
// JOB DEFLIBR
// DLBL DGIWORK,'SDF2.WORK.LIB',99/365,SD
// EXTENT ,SDFLB1,1,0,1620,60
// EXEC LIBR
 DEFINE L=DGIWORK
 DEFINE S=DGIWORK.DGI
/*
/&
* $$ EOJ
```

Defining object and autosave libraries and sublibraries

Define object libraries and sublibraries as required for your installation. You can freely choose the names of the sublibraries.

The size of a library depends on the number and size of the objects you want to store in it. As a rule of thumb, assume 15 KB (15 library blocks) as the average size of an object. This should be a reasonable value for your library size estimates.

When device lists are used and data structures are generated, the number of generated objects may be a multiple of the number of SDF II internal objects.

If you do not have the figures to calculate the space requirements, you may prefer to define the libraries in VSAM space with primary and secondary allocation values.

To use the autosave feature of SDF II, define a sublibrary to hold the autosave data. The name of the sublibrary can be freely chosen and must be specified by the user in the Specify Libraries dialog of SDF II.

Customizing the SDF II start-up jobs

This section explains how to customize the SDF II start-up jobs for full-screen operation and for batch operation.

Customizing the full-screen start-up job

The SDF2 transaction creates VSE/POWER statements and job control statements dynamically for the SDF II job. The exit routine DGIJOBFS contains the job skeleton, the defaults for the substitution variables, and the messages used by the CICS/VSE transaction.

Figure 4 shows the IBM-supplied start-up job skeleton.

```
* $$ JOB JNM=&JOBNAME,CLASS=&JOBCLAS,DISP=D,SEC=(&USERID,&PWD)
* $$ LST CLASS=&LSTCLAS,DISP=&LSTDISP
* $$ PUN CLASS=&PUNCLAS,DISP=&PUNDISP
// JOB &JOBNAME &ACCOUNT
// LIBDEF PROC,SEARCH=IJSYSRS.SYSLIB
// OPTION &OPTION
// EXEC PROC=DGIPROC,                                                 *
               XPCCID=&XPCCID,                                        *
               REQU=&REQU,                                            *
               LANG=&LANG,                                            *
               SPEC=&SPEC
/&
* $$ EOJ
```

Figure 4. IBM-supplied default start-up job skeleton

The substitution variables (prefixed with &) in the job skeleton resolve to the defaults and the options specified in the SDF II invocation. You can modify the skeleton and the defaults to meet your installation's needs.

Attention: Do not change the parameters in the // EXEC PROC=DGIPROC statement.

The LIBDEF PROC must ensure that the customized DGIPROC.PROC is called.

To modify the exit:

1. Change the source of DGIJOBFS.
2. Assemble and link-edit the new DGIJOBFS. See member DGIJOBFS.A for details.

   To assemble and link-edit DGIJOBFS, SDF II provides two sample jobs: DGIJOBF.Z and DGIJOBF2.Z. Use DGIJOBF2 if you have added EXEC CICS statements to the DGIJOBFS source.
3. Activate the changed DGIJOBFS module, using the CICS/VSE Master Terminal Transaction. Enter the following:

   cemt set program(dgijobfs) newcopy

Customizing the batch start-up job

Modify the following sample job skeleton to meet the needs of your installation. It is in DGIIJBAT.Z in the SDF II installation library.

```
* $$ JOB JNM=DGIIJBAT
// JOB DGIIJBAT
/. C -----------------------------------------------------------------
/. C     SCREEN  DEFINITION  FACILITY II  -  RELEASE 6 LEVEL 0
/. C          LICENSED MATERIAL - PROGRAM  PROPERTY OF IBM
/. C 5746-XXT (C) Copyright IBM Corp. 1979, 1997.  ALL RIGHTS RESERVED.
/. C Use a libdef proc to ensure the customized DGIPROC.PROC is called.
/. C -----------------------------------------------------------------
// LIBDEF PROC,SEARCH=IJSYSRS.SYSLIB
// EXEC PROC=DGIPROC,REQU='',SPEC='',LANG=''

put any SYSIPT data here

/*
/&
* $$ EOJ
```

For information about how to run an SDF II batch job, see “Starting SDF II in batch mode” on page 22.

Customizing the invocation procedure

To set up the environment of SDF II to suit the needs of your organization, use the procedure DGIPROC.PROC. DGIPROC.PROC is called for full-screen and batch operation.

```
// PROC XPCCID='',REQU='',LANG='',SPEC=''
/. C ------------------------------------------------------------------
/. C ******************* SDF II Startup Procedure *********************
/. C     SCREEN  DEFINITION  FACILITY II  -  RELEASE 6 LEVEL 0
/. C          LICENSED MATERIAL - PROGRAM  PROPERTY OF IBM
/. C 5746-XXT (C) Copyright IBM Corp. 1979, 1997.  ALL RIGHTS RESERVED.
/. C
/. C   The DGIPROC procedure defines the JCL environment for the
/. C   SDF II invocation in both interactive and batch mode.
/. C
/. C   The procedure should contain DLBL/EXTENT statements for all
/. C   libraries that may be used by SDF II, LIBDEF statements as
/. C   described below, the invocation of the SDF II stub DGISTART,
/. C   and, of course, any additional statements as required by your
/. C   shop.
/. C
/. C   For more information how to customize DGIPROC, refer to the
/. C   SDF II Administration Guide, SH12-6311.
/. C
/. C   Note: SDF II object libraries that are NOT used to hold
/. C         generated prototype objects (panels and execs) need not
/. C         to be added to the LIBDEF SEARCH chain.
/. C
/. C         DLBL/EXTENT information however must be available for
/. C         ALL object libraries.
/. C
/. C ------------------------------------------------------------------
/. C
/. C As this procedure generally will contain LIBDEF * and LIBDEF PROC
/. C statements, special restrictions apply:
/. C     - it must reside in IJSYSRS.SYSLIB
/. C     - after the LIBDEF, it must not call another procedure
/. C     - it must not be nested
/. C
/. C For more information refer to VSE/ESA System Control Statements.
/. C
/. C ------------------------------------------------------------------
/. C Enter DLBL/EXTENT statements for SDF II object libraries
/. C (unless available in the standard labels).
/. C ------------------------------------------------------------------
// DLBL SDFOBJ,'SDF2.OBJECTS.LIB'
// EXTENT ,SDFLB2
/. C ------------------------------------------------------------------
/. C Invocation parameter SPEC may be used for any additional job
/. C tailoring.
/. C ------------------------------------------------------------------
// IF SPEC='' THEN
// GOTO SETLIB
/. C Add your additional job tailoring statements here.
/. SETLIB
/. C ------------------------------------------------------------------
/. C Define the search chain libraries.
/. C Specify a null string for the libraries that are not used in
/. C your installation, or that are, with the desired position,
/. C in the permanent search chain.
/. C ------------------------------------------------------------------
/. C The DGIWORK library is always the chain's head.
// SETPARM WORK='DGIWORK.DGI'
/. C ------------------------------------------------------------------
/. C Next, with leading comma, all PROTOTYPE OBJECT libraries.
// SETPARM PROTO=''
/. C ------------------------------------------------------------------
/. C Next, with leading comma, any SDF II customization libraries.
// SETPARM CONFIG=',PRD2.CONFIG'
```

```
/. C ------------------------------------------------------------------
/. C Next, with leading comma, any NLS feature install libraries.
/. C The sample assumes that NLS features are installed in library
/. C PRD2.DGIxxx, where xxx is the 3-character language code.
/. C ------------------------------------------------------------------
// IF LANG¬='' THEN
// SETPARM LANG=',PRD2.DGI&LANG'
/. C ------------------------------------------------------------------
/. C Last, but not least, the SDF II base product install library.
// SETPARM INST=',PRD2.PROD'
/. C ------------------------------------------------------------------
/. C Build LIBDEF SEARCH statement and do not change the sequence.
// LIBDEF *,SEARCH=(&WORK.&PROTO.&CONFIG.&LANG.&INST.)
/. C ------------------------------------------------------------------
/. C If there is no XPCCID, we are running in batch.
/. C ------------------------------------------------------------------
// IF XPCCID='' THEN
// GOTO BATCH
/. C ------------------------------------------------------------------
/. C If called from CICS, a LIBDEF PHASE,CATALOG statement must
/. C denote the user profile library.
/. C The EXEC statement must not be changed.
/. C ------------------------------------------------------------------
// LIBDEF PHASE,CATALOG=PRD2.CONFIG
// EXEC DGISTART,SIZE=DGISTART,PARM='XPCCID=&XPCCID,R,DGISETUP &REQU SP*
               EC(&SPEC)'
// GOTO END
/. BATCH
/. C ------------------------------------------------------------------
/. C Here for SDF II batch invocation, without // LIBDEF PHASE,CATALOG.
/. C The EXEC statement must not be changed
/. C ------------------------------------------------------------------
// EXEC DGISTART,SIZE=DGISTART,PARM='SELECT=DGISETUP &REQU SPEC(&SPEC)'
/. END OF PROCEDURE
```

Notes:

1. Do not change the // EXEC DGISTART statement.
2. Do not change the contents of the variable XPCCID. This variable is used only for SDF II internal purposes.
3. Add to the DGIPROC.PROC the DLBL and EXTENT information for all object libraries, unless it is already available in the standard labels.
4. Include LIBDEF statements. The LIBDEF *,SEARCH statement must ensure that:
   - The DGIWORK.DGI library is in the first position
   - All sublibraries that contain prototype objects and EXECs are included
   - If phases are to be called by your prototype EXECs, their sublibraries are included
   - Any customized SDF II phases and tables are found before the IBM-supplied members in the SDF II installation libraries

     Only those SDF II object libraries that are used for prototyping need be included in the LIBDEF search chain.
5. For full-screen operation only, include a // LIBDEF CATALOG statement for the library that contains the user profiles.

The catalog library is required by the dialog management services to store user profiles for full-screen operation. In batch operation, no // LIBDEF CATALOG must be active.

Copying the SDF II start-up procedure to the system library

If your customized DGIPROC.PROC contains a // LIBDEF PROC or // LIBDEF * statement, DGIPROC.PROC must fulfill the following criteria:

- It must reside in the system sublibrary IJSYSRS.SYSLIB
- It must not be nested
- It must not call another procedure after the LIBDEF statement

For more information refer to *VSE/ESA System Control Statements*.

Use VSE/ESA Librarian to copy DGIPROC.PROC into IJSYSRS.SYSLIB. Modify the following job control statements to meet the requirements of your installation. If you installed SDF II in a library other than PRD2.PROD, modify the job accordingly.

```
* $$ JOB JNM=COPYSET
// JOB COPYSET
// EXEC LIBR
 CONN S=PRD2.PROD:IJSYSRS.SYSLIB
 COPY DGIPROC.PROC REPLACE=YES
/*
/&
* $$ EOJ
```

Setting up dynamic partitions

SDF II requires a batch partition (either static or dynamic) for each interactive user. Dynamic partitions are better for full-screen SDF II execution, because they provide an identical environment for all users. The partition size should be 5 MB.

In accordance with the default dynamic class table shipped with VSE/ESA, SDF II is initially set up to use CLASS=Z, which by default provides for only one partition. To enable multiple dynamic partitions of the recommended size, modify the dynamic class table DTR$DYNC. For information on how to modify the dynamic class table, refer to *VSE/POWER Administration and Operation*.

Sample entry in the dynamic class table

The MAX-P value is defined to provide for five concurrent users of SDF II.

```
          CLASS  ALLOC   SIZE   SP-GETV LUBS PROFILE  MAX-P   ENABLED
----------+------++++----+++++--+++++---+++--++++++++-++------+-----------------
CLASS=    Z         5     1024    516    50  STDPROF   5      Y
          POWER SPOOLED DEVICES
----------+++-+++-+++-+++-+++-+++-+++-+++-+++-+++-+++-+++-+++-+++---------------
READER=   FEC
PRINTERS= FEE
PUNCHES=  FED
------------------------------------------------------------------------
```

Note: If you define another class for use with SDF II, you can also change the default in the full-screen invocation skeleton DGIJOBFS.

Changing the translation tables

There are three translation tables:

DGIYTUC Translates characters to uppercase
DGIYTLC Translates characters to lowercase
DGIYTDC Translates nondisplayable characters to displayable characters

In some SDF II dialogs, you can use these line commands:

u For translation to uppercase
l For translation to lowercase

In the uppercase translation table, you find for each hexadecimal code the corresponding hexadecimal code that is used when you issue the **u** line command.

Figure 5 and Figure 6 on page 16 show the assembler code of the tables that are used for this translation.

```
************************************************************************
* DGIYTUC - TRANSLATE TO UPPER CASE TRANSLATE TABLE
************************************************************************
DGIYTUC  CSECT ,  0 1 2 3 4 5 6 7 8 9 A B C D E F
         DC    X'000102030405060708090A0B0C0D0E0F'    H'0x'
         DC    X'101112131415161718191A1B1C1D1E1F'    H'1x'
         DC    X'202122232425262728292A2B2C2D2E2F'    H'2x'
         DC    X'303132333435363738393A3B3C3D3E3F'    H'3x'
         DC    X'404142434445464748494A4B4C4D4E4F'    H'4x'
         DC    X'505152535455565758595A5B5C5D5E5F'    H'5x'
         DC    X'606162636465666768696A6B6C6D6E6F'    H'6x'
         DC    X'707172737475767778797A7B7C7D7E7F'    H'7x'
         DC    X'80C1C2C3C4C5C6C7C8C98A8B8C8D8E8F'    H'8x'
         DC    X'90D1D2D3D4D5D6D7D8D99A9B9C9D9E9F'    H'9x'
         DC    X'A0A1E2E3E4E5E6E7E8E9AAABACADAEAF'    H'Ax'
         DC    X'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF'    H'Bx'
         DC    X'C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF'    H'Cx'
         DC    X'D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF'    H'Dx'
         DC    X'E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF'    H'Ex'
         DC    X'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'    H'Fx'
         END   DGIYTUC
```

Figure 5. Table for translation to uppercase

For example, to find the uppercase value for hexadecimal code 81, look along the line marked 8 on the right of the table. Then look down the column marked 1 on the top of the table. Where line 8 and row 1 intersect, you find the value C1.

You read the lowercase translation table in the same way.

```
***********************************************************************
* DGIYTLC - TRANSLATE TO LOWER CASE TRANSLATE TABLE
***********************************************************************
DGIYTLC  CSECT ,  0 1 2 3 4 5 6 7 8 9 A B C D E F
         DC    X'000102030405060708090A0B0C0D0E0F'    X'0x'
         DC    X'101112131415161718191A1B1C1D1E1F'    X'1x'
         DC    X'202122232425262728292A2B2C2D2E2F'    X'2x'
         DC    X'303132333435363738393A3B3C3D3E3F'    X'3x'
         DC    X'404142434445464748494A4B4C4D4E4F'    X'4x'
         DC    X'505152535455565758595A5B5C5D5E5F'    X'5x'
         DC    X'606162636465666768696A6B6C6D6E6F'    X'6x'
         DC    X'707172737475767778797A7B7C7D7E7F'    X'7x'
         DC    X'808182838485868788898A8B8C8D8E8F'    X'8x'
         DC    X'909192939495969798999A9B9C9D9E9F'    X'9x'
         DC    X'A0A1A2A3A4A5A6A7A8A9AAABACADAEAF'    X'Ax'
         DC    X'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF'    X'Bx'
         DC    X'C0818283848586878889CACBCCCDCECF'    X'Cx'
         DC    X'D0919293949596979899DADBDCDDDEDF'    X'Dx'
         DC    X'E0E1A2A3A4A5A6A7A8A9EAEBECEDEEEF'    X'Ex'
         DC    X'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'    X'Fx'
         END   DGIYTLC
```

Figure 6. Table for translation to lowercase

Figure 7 shows the assembler code of the table that translates nondisplayable codes. For example, it translates printer and IBM 3270 control codes to periods.

```
***********************************************************************
* DGIYTDC - TRANSLATION TABLE FOR DISPLAYABLE SCREEN CHARACTERS
***********************************************************************
DGIYTDC  CSECT ,  0 1 2 3 4 5 6 7 8 9 A B C D E F
         DC    X'004B4B4B4B4B4B4B4B4B4B4B4B0E0F'    X'0x'
         DC    X'4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B'    X'1x'
         DC    X'4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B'    X'2x'
         DC    X'4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B'    X'3x'
         DC    X'404142434445464748494A4B4C4D4E4F'    X'4x'
         DC    X'505152535455565758595A5B5C5D5E5F'    X'5x'
         DC    X'606162636465666768696A6B6C6D6E6F'    X'6x'
         DC    X'707172737475767778797A7B7C7D7E7F'    X'7x'
         DC    X'808182838485868788898A8B8C8D8E8F'    X'8x'
         DC    X'909192939495969798999A9B9C9D9E9F'    X'9x'
         DC    X'A0A1A2A3A4A5A6A7A8A9AAABACADAEAF'    X'Ax'
         DC    X'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF'    X'Bx'
         DC    X'C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF'    X'Cx'
         DC    X'D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF'    X'Dx'
         DC    X'E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF'    X'Ex'
         DC    X'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'    X'Fx'
         END   DGIYTDC
```

Figure 7. Table for translation to displayable screen characters

To modify the translation tables to meet your needs:

1. Retrieve the members DGIYTUC.A, DGIYTLC.A, and DGIYTDC.A from the SDF II installation library.
2. Adapt the contents of the tables, as appropriate.
3. Assemble the tables, catalog the object decks in the SDF II installation library, and link the affected SDF II phases. Member DGIKJTRN.Z provides a sample assemble, catalog, and link-edit job.

Notes:

1. Always keep copies of the original tables, in case you need to use the original SDF II defaults.
2. Create a backup copy of the customized tables, because any program maintenance may change them.

Destination of print output

Where print output is sent depends on its character set and, for DBCS print output, whether it uses field outlining:

- Normal print output (SBCS) is sent to SYSLST.
- DBCS print output without field outlining is sent to SYS021.
- DBCS print output with field outlining is sent to SYS022.

To enable DBCS printing for your installation, add the appropriate // ASSGN statements to your DGIPROC.PROC procedure.

Verifying the SDF II installation

To verify that SDF II is correctly installed, work through this test procedure:

1. Start SDF II:
   1. a. Start SDF II as a CICS/VSE transaction, **sdf2**.
   2. b. On the Select an SDF II Main Function panel, select option **1**, SDF II Functions.
   3. c. On the Select an SDF II Function panel, select option **8**, Specify Libraries.
2. Specify your library:
   1. a. On the Specify Libraries panel:
      1. 1) In the **Library name** field of the library **1** row, type the name of the object library you defined in “Defining object and autosave libraries and sublibraries” on page 10.
      2. 2) In the **Search for related objects** field, type **1**
   2. b. Press the **End** key (PF3) to return to the Select an SDF II Function panel.

3. Import an object:

   a. On the Select an SDF II Function panel, enter **9.2** to display the Specify Import Utility Parameters panel.

   b. Import the SDF II panel, as follows:

      Type Option **3**, ISPF panel.

      Library Type the VSE library name of the SDF II installation library. (The default name is PRD2.PROD.)

      Member **dgiiu000.a**

      Target library Type **1** to identify the object library.

   c. Press the **Enter** key to process the request. SDF II imports the object into the specified library. You should receive no error messages.

   d. Press the **End** key (PF3) three times to return to the Select an SDF II Function panel.

4. Edit the object:

   a. On the command line, enter **7** to display the Define Object List panel.

   b. Press **Enter** to list your panels.

   c. On the List Objects panel, enter **e1** in the line command area beside the entry for the ISPF panel DGIIU000. The Define Panel Characteristics panel is displayed.

   d. Type a description for the panel.

   e. Press the **End** key (PF3) twice, to display the Exit Editor window.

   f. Select **Save**, to save the panel and return to the List Objects panel.

5. Generate the object:

   a. In the line command area beside the panel name, type **g** and, in the **Operands** field, type **1**, to specify generation for CICS/BMS.

   b. Press the **Enter** key. SDF II displays the Specify Generation Parameters panel.

   c. On the Specify Generation Parameters panel, select **Generate data structure** and **Generate CICS/BMS macros**, and specify the following:

      Data structure output: Sublibrary name
      Type the name of the object library you defined in "Defining object and autosave libraries and sublibraries" on page 10.

      Data structure output: Member type
      Type **d**

      Data structure output: User exit
      Specify either of the following user exit routines to browse the output:

      DGICXBRW Specify this user exit routine if DITTO/ESA is not installed on your system in accordance with the requirements stated in the *SDF II Program Directory*.

      DGICXBUD Specify this user exit routine only if DITTO/ESA is installed on your system in accordance with the requirements stated in the *SDF II Program Directory*.

CICS/BMS macro output: Sublibrary name
Type the ID of the object library you defined in “Defining object and autosave libraries and sublibraries.”

Data structure output: Member type
Type **a**

CICS/BMS macro output: User exit
Specify the output browser user exit routine, as you did in step 5c on page 18.

d. Press the **Enter** key to process the request. SDF II generates the object. The generated data structure is displayed.

e. Press the **End** key (PF3) to display the BMS macros.

f. Press the **End** key again to display the List Generation Messages panel. There should be no error messages.

g. Type **/** in the **Print messages** field.

h. Press the **End** key (PF3) to leave the List Generation Messages panel and send print output to SYSLST. The Specify Generation Parameters panel is displayed.

i. Press the **End** key (PF3) to return to the List Objects panel. The message `*Generated` is displayed in the **Operands** field.

6. Print the object:

   a. In the line command area of the DGIIU000 entry, enter **p**. SDF II processes the request, then displays the message `*Printed` in the **Operands** field.

   b. Press the **End** key (PF3) three times to return to the Select an SDF II Main Function panel.

This ends the SDF II installation verification test.

Chapter 3. Starting SDF II

This chapter explains how to start SDF II in full-screen mode and in batch mode.

Starting SDF II in full-screen mode

SDF II is started and controlled by a CICS transaction and runs in a dynamic or static VSE/ESA partition.

On a CICS/VSE terminal, invoke the SDF2 transaction with all defaults or with override parameters. For example, you could invoke SDF II in any of the following ways:

sdf2
Displays the Select an SDF II Main Function panel

sdf2 proto1
Invokes the saved prototype EXEC PROTO1

sdf2 1.1 (user=didusr01 pwd=x3aa2pq
Displays the Identify Panel panel of the panel editor, and overrides the user ID and password

sdf2 (jobclass=z lstdisp=d touser=puchas
Overrides some POWER job defaults

sdf2 (option=partdump,nosysdump
Overrides some JCL defaults

Syntax

SDF2 [*request*] [**(** [*options*]]

request is:

[{ *menu-option* |
FUNCTIONS |
PROTOTYPE |
CUSTOMIZE |
exec [*parm*] }]

options is:

[[**ACCOUNT=***account*]
[**JOBCLASS=***jobclass*]
[**LANG=***language*]
[**LSTCLASS=***list-class*]
[**LSTDISP=***list-disposition*]
[**OPTION=***option*]
[**PUNCLASS=***punch-class*]
[**PUNDISP=***punch-disposition*]
[**PURGE**]
[**PWD=***password*]
[**SPEC=***specification*]
[**TOUSER=***to-userid*]
[**USER=***userid*]
[**WAIT=***wait*]]

where:

menu-option An option for a selection panel, such as **1** or **2.3**.

FUNCTIONS To bypass the primary panel and immediately enter the editors and utilities dialog.

PROTOTYPE To bypass the primary panel and immediately enter the run prototype dialog.

CUSTOMIZE To bypass the primary panel and immediately enter the customization dialog.

exec The name of a REXX EXEC, such as the name of a generated prototype. For more information about how to run a prototype, refer to *SDF II Run-Time Services*.

parm Parameters for the REXX EXEC.

ACCOUNT=*account* The job account information.

JOBCLASS=*jobclass* The job class for SDF II.

LANG=*language* The 3-character identifier of the national language to be used, such as ENU for U.S. English.

The default language is determined from the library search order, which you set in DGIPROC.PROC.

LSTCLASS=*list-class* The class for SDF II print output.

LSTDISP=*list-disposition* The disposition for SDF II print output.

OPTION=*options* The parameters for the OPTION JCL statement separated by commas (no blanks are allowed).

PUNCLASS=*punch-class* The class for SDF II punch output.

PUNDISP=*punch-disposition* The disposition for SDF II punch output.

PURGE To delete the joblog and printed output at the end of the SDF II session.

PWD=*password* The password.

SPEC=*specification* A customizable specification. In this field, you can specify a string of up to 8 characters for use in the start-up procedure.

TOUSER=*to-userid* The destination user ID for SDF II print output.

USER=*userid* The user ID.

WAIT=*wait* The timeout value, in seconds, to limit waiting for connection to the SDF II partition. The range is 1 through 9999; the default is 120.

Starting SDF II in batch mode

For batch operation, you use a REXX EXEC to pass the invocation parameters to SDF II. This start-up EXEC can be either the IBM-supplied DGIIX EXEC or one that you provide.

For information about how to write a start-up EXEC, see “Writing a start-up REXX EXEC” on page 58.

You can perform the following SDF II tasks in a batch environment:

- Generate objects
- Import objects
- Print objects
- Print the online reference
- Construct panels
- Extract and modify panel data
- Copy and delete objects

Starting SDF II with the IBM-supplied start-up EXEC DGIIX

For the DGIIX start-up EXEC, you specify your invocation parameters in a request file, rather than by typing it into the panel input fields. The request file can be a member of a VSE sublibrary, or it can be added to the SDF II job as SYSIPT data.

In a batch SDF II job that has been customized for your installation, modify the request parameter:

`// EXEC PROC=DGIPROC,REQU=''`
Specifies a request file from SYSIPT.

`// EXEC PROC=DGIPROC,REQU='DGIIX` *`lib.sublib member.type`*`'`
Specifies the start-up EXEC DGIIX, and the request file.

Any SYSIPT data must immediately follow the DGIPROC.PROC procedure invocation and must be delimited by /*. See “Preparing a request file for DGIIX” on page 40 for information on how to write a request file.

The syntax of the IBM-supplied batch job sample is as follows:

```
* $$ JOB JNM=DGIIJBAT
// JOB DGIIJBAT
   ⋮
// LIBDEF PROC,SEARCH=IJSYSRS.SYSLIB
// EXEC PROC=DGIPROC,REQU='request',LANG='language',SPEC='specification'

put your SYSIPT parameters here

/*
/&
* $$ EOJ
```

where:

request
Specifies the start-up EXEC and the parameter string to be passed to it. For DGIIX, the parameter string identifies the request file. If you do not specify a request parameter, DGIIX is called and reads the request data from SYSIPT.

language
Specifies the 3-character identifier of the national language to be used, such as ENU for U.S. English.

The default language is determined from the library search order, which you set in DGIPROC.PROC.

You can use the LANG parameter to customize the LIBDEF search chain in your DGIPROC.PROC procedure.

specification
Is an additional parameter that you can use to control the processing of the DGIPROC procedure.

Chapter 4. Customizing SDF II profiles and tables

You can customize various parts of SDF II, such as:

- Defining defaults, device types, emphasis classes, and CUA elements
- Adapting print utility skeletons
- Changing the SDF II keylists
- Making changes available

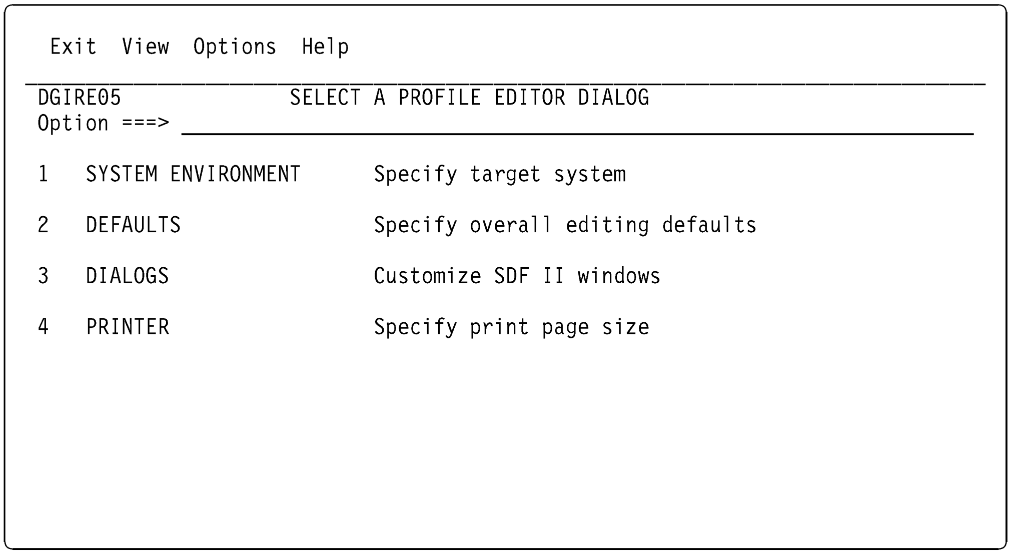
For most of these customization tasks, you can use SDF II dialogs.

This chapter explains these SDF II administrator tasks. For more information, refer to the online reference.

Defining installation defaults

To define the installation defaults for SDF II, first

1. Start SDF II as a CICS/VSE transaction, **sdf2**.
2. On the Select a Main SDF II Function panel, enter **1**, to display the Select an SDF II Function panel.
3. Enter **10**, to display the Select a Profile Editor Dialog panel.



```
  Exit  View  Options  Help
 ______________________________________________________________________
 DGIRE05              SELECT A PROFILE EDITOR DIALOG
 Option ===> __________________________________________________________

 1   SYSTEM ENVIRONMENT      Specify target system

 2   DEFAULTS                Specify overall editing defaults

 3   DIALOGS                 Customize SDF II windows

 4   PRINTER                 Specify print page size
```

Specify target system
In this dialog, you define the target system. The target system can be CICS/BMS, IMS/MFS, ISPF, GDDM–IMD, CSP/AD, or VisualGen.

Specify overall editing defaults
In this dialog, you can specify various defaults:

- Defaults for all SDF II editors:

  Autosave
  Specifies whether and how often an edited object is to be saved automatically in the autosave library.

Confirm save on exit
Specifies whether users are to be asked to confirm that the object is to be saved.

Line numbers
Turns on or off the line numbers of window panels.

- Defaults for the panel editor:

Preserve
Specifies whether the data structure is to be protected during editing.

Capitals
Controls the translation of lowercase letters to uppercase letters in the Format window.

Nulls
Controls whether trailing nulls or blanks fill each line of the Format window of the panel editor.

Line commands
Controls whether the line command area is to be shown in the Format window.

Mixed-case names
Specifies whether data structure names can be specified in mixed case.

Marks
Defines the characters to be used as marks when creating a new panel.

- Confirmation of delete command:

List objects
Specifies whether users are to be asked to confirm a **d** (delete) line command in the List Objects window.

Define fields
Specifies whether users are to be asked to confirm a **d** (delete) line command in the Define Fields dialog.

Customize SDF II windows
Use the Customize a Window panel to change the layout of the selected window. You can choose the columns displayed in the dialog and the width of each column. Changes in the layout of the window apply only to the target system defined in the profile editor. You can define different layouts of the same window for each target system.

Specify print page size
In this dialog you define the page size used by the print utility.

If you change the page width and want to print complete objects (not only the formats), you need to adjust the skeletons that define the print layout. For details on how to do this, see “Adapting print utility skeletons” on page 29.

Making changed defaults available

The changes you make are saved in *userid*.PROFDGI1. How to give other users access to this profile is explained in “Making changes to the user profile available” on page 39.

Specifying libraries

In the Specify Libraries dialog, which is option 8 on the Select an SDF II Function panel, specify the object and autosave libraries you want to use in SDF II.

The changes you make are saved in *userid*.PROFDGI1. How to give other users access to this table is explained in "Making changes to the user profile available" on page 39.

Defining device types

The device table distributed with SDF II lists all the types of devices supported by SDF/CICS, CICS/BMS, IMS/MFS, GDDM–IMD, CSP/AD, and VisualGen. In addition, you can define new device types or device lists that are suitable for your installation.

In SDF II, device types can be names of devices or names of device lists. A device name corresponds to a single type of device supported by the target systems. A device list contains the names of a set of devices that have the same characteristics.

The SDF II functions do not distinguish between devices and device lists. Wherever you are required to enter the name of a device type, you can enter the name of either a device or a device list.

When necessary, such as for generation, each device in the device list is processed separately. Generation, for example, is thus performed for each device in the list.

Defining devices

To enter the SDF II device type editor:

1. On the Select an SDF II Main Function panel, select option **3**, SDF II Customization. The Select an SDF II Customization Dialog panel is displayed.
2. Select option **1**, Device Table, to display the Define Devices panel, where you can find and specify:
   - The device name
   - An indication as to whether this is a device list
   - Entries that show the corresponding device code in the target systems
   - A column with a comment

These line commands are available on the Define Devices panel:

c Copy device type entries
d Delete device type entries
i Insert new devices
s Select and display:

- The Define Device List panel for a device list. There you define the devices that belong to the device list.
- The Define Device Type Characteristics panel for single devices. There you define the characteristics of the device. You cannot change the characteristics of the devices in the list provided by IBM.

Make sure that the definition of the device type contains an entry for each target system for which the device type may be used. Otherwise you cannot create or generate an object for that device type and target system.

The import utility uses the first device type that matches the device characteristics of the object to be imported as the device type of the imported object. Consider this matching algorithm when you add or change entries in the device table.

- The CICS/BMS device suffixes for target systems in SDF II may differ from the device suffixes to be imported. If this is the case, an entry specifying the correct suffix needs to be added to the device table to import the object successfully. Suffixes of device entries provided by IBM cannot be changed.
- If you used values other than the default values in the device characteristics table (DVDDCT) of SDF/CICS, modify the SDF/CICS device code (column SDF) in the device table.
- Because the import utility takes the first matching entry in the device table, the resulting device type may be different in size or supported attributes from the size or attributes of the object to be imported.

  To avoid the loss of attributes, you may need to replace the entry in the target system column with blanks (for IMS/MFS and SDF/CICS this column can be modified) or to add your own device type to the device table before the first matching device type provided by IBM.

Defining a device list

A device list contains a set of devices that have the same characteristics. It cannot contain another device list. Define a device in the SDF II device table, then include it in a device list. A device list can contain up to 24 devices.

The device characteristics of a device list are the superset of the characteristics of the devices in the list. But all device types contained in the device list must have the same setting for the device attribute ADJACENT FIELDS.

A device list for IMS/MFS should contain only devices of the IBM 3270 display family or the IBM 3270 printer devices.

Note: It is recommended that only devices with the same display size or page size be grouped into the same device list.

Defining device type characteristics

On the Define Device Type Characteristics panel you can define the following characteristics:

- Device dimensions: depth, width, and an indication of whether the device size can be extended
- Character cell size (vertical and horizontal)
- Information about partitioning
- Device attributes (adjacent fields, numeric lock feature, light pen, number of PF keys)
- Input, output
- Extended device attributes (such as extended colors and extended highlighting)
- Device features (such as tabulators)

Making changed device types available

The changes you make are saved in DGIKFDT.A and DGIKF10.A. How to give other users access to these tables is explained in “Making changes to SDF2/DM tables available” on page 39.

Defining emphasis classes

You can combine presentation attributes into groups called emphasis classes. Each emphasis class is identified by a 2-byte name.

Emphasis classes provide a convenient means to change the attributes of screen fields throughout the installation, without having to re-edit the panels: you need only regenerate them.

You define emphasis classes in the customization dialogs.

1. To start the customization dialogs, select option **3**, SDF II Customization, on the Select an SDF II Main Function panel. The Select SDF II Customization Dialog panel is displayed.
2. On the Select SDF II Customization Dialog panel, select option **3** to display the Define Emphasis Class panel. That panel lists the previously defined emphasis classes. On it you can specify:

   - The name of the emphasis class
   - The list of attributes included
   - A comment

   You can create, delete, and modify emphasis classes by using the following line commands, which are available on the Define Emphasis Class panel:

c	Copy emphasis class entries
d	Delete emphasis class entries
i	Insert new entries
r	Repeat an entry

   You can define emphasis classes for fields, marks, and attribute descriptors.

   You cannot delete or rename the emphasis classes provided by IBM, because the SDF II panel construction utility uses them.

To use an emphasis class, specify its 2-byte name, instead of single attributes, in the panel editor. The syntax is:

CLASS *cc*

where *cc* is the name of the emphasis class.

When SDF II generates or tests a panel, it replaces any defined emphasis classes with the attributes they represent.

Making changed emphasis classes available

The changes you make are saved in DGIKF30.A. How to give other users access to this table is explained in “Making changes to SDF2/DM tables available” on page 39.

Changing CUA panel element attributes

The CUA panel element types are predefined in SDF II. You can, however, change the attributes for each CUA panel element type.

To make these changes:

1. On the Select an SDF II Main Function panel, select option **3**, to display the Select an SDF II Customization Dialog panel.
2. Select option **4**, CUA Attributes, to display the Edit CUA Attributes panel. This panel lists the CUA panel element types with their current attributes and comments. The protection attribute is displayed, but cannot be changed. On this panel you can specify or change the attributes for each CUA panel element type.

   You can make any line the first line by using the **/** line command. No other line commands are available on the Edit CUA Attributes panel.

When SDF II generates a panel for ISPF, it generates the corresponding CUA attributes. For any other target system, it replaces any defined CUA attributes with the equivalent attributes appropriate to that target system.

Making changed attributes available

The changes you make are saved in DGIKF40.A. How to give other users access to this table is explained in "Making changes to SDF2/DM tables available" on page 39.

Adapting print utility skeletons

The print utility of SDF II uses skeletons to produce the printed output. These print utility skeletons are in the SDF II installation library as *membername*.A. (For member names, see Figure 8 on page 30.)

During the first pass of the print utility, file-tailoring services are used to create a temporary file, the print input file, from the appropriate print utility skeletons. Variables and tables are replaced by their actual contents. In the second pass of the print utility, SDF II uses this print input file to create the printed output.

Why to adapt

You may need to adapt print utility skeletons because:

- You want to print the message list on a different printer.

  Change the general control record in DGIYSMS.A (see "General control record" on page 32). For example, if you change it to:

  ```
  G &IYVMSPAG 120 60 4
  ```

  the output will be routed to a DBCS printer.

- The print width of your printer is different from the default width. The print utility skeletons are prepared for a print width of 120 characters.

Because the record length of the print input file is 80 bytes, you may need continuation lines. A logical line (variables replaced by their values and continuation lines concatenated) must not exceed the defined width. If any lines exceed the defined width, they are truncated and you get an error message after printing.

Before you change the print width in the print utility skeletons, you must change the print width using the profile editor.

Formats are adapted automatically to a different print width. You do not need to change the skeletons.

If you want to print other parts of an object, you have to adapt the skeletons.

Note: Always keep copies of the original print utility skeletons in case you want to use the original SDF II settings.

Where to adapt

When the SDF II print utility creates the print input file, it includes print utility skeletons appropriate for the kind of print request.

Figure 8 explains where to find the information you may need to adapt print utility skeletons to your requirements.

Figure 8 (Part 1 of 2). Skeleton members

For printing ...	You can change ...	In skeleton ...
Any object	General control records General header settings	DGIUS1IG.A
Panel groups	Panel list	DGIUS1G1.A
	Panel group characteristics	DGIUS1G2.A
	BMS characteristics	DGIUS1G3.A
	Programmed symbol set list	DGIUS1G4.A
	Layout	DGIUS1G5.A
Panels	Format	DGIUS1P1.A
	General characteristics	DGIUS1P2.A
	GDDM characteristics	DGIUS1P3.A
	BMS characteristics	DGIUS1P4.A
	Action bar choices	DGIUS1PA.A
	Fields	DGIUS1P5.A
	Attributes	DGIUS1P6.A
	ISPF characteristics	DGIUS1P7.A
	CSP or VisualGen characteristics	DGIUS1P8.A
	Editing characteristics	DGIUS1P9.A
	Structure	DGIUS1PB.A
	MFS format set	DGIUS1PC.A
	MFS characteristics / structure	DGIUS1PX.A
	Format for MFS stream devices	DGIUS1PZ.A
AID table	AID table	DGIUS1A1.A

Figure 8 (Part 2 of 2). Skeleton members

For printing ...	You can change ...	In skeleton ...
Control table	Control table	DGIUS1B1.A
Partition sets	Partition set characteristics	DGIUS1S1.A
	Partition list	DGIUS1S2.A
	Layout	DGIUS1S3.A
Prototype table	Prototype table	DGIUS1W1.A
Message list	General controls	DGIYSMS.A
Online reference	Headings	DGIUS4HD.A
	Index	DGIUS4IX.A
	Body	DGIUS4TB.A
	Table of contents	DGIUS4TC.A
	Back matter	DGIUS40B.A
	Front matter	DGIUS40F.A

What to adapt

In the print utility skeletons, you can change lines or add new lines if necessary. The lines of the print utility skeletons are:

File-tailoring statements, which are the lines that start with a right parenthesis [)].

Print utility records, which are all the other nonblank lines in the skeletons.

Note: Any blank data records in the input data are deleted from the file-tailoring output.

File-tailoring statements and print utility records can contain the names of SDF2/DM variables and table elements, which are prefixed by an ampersand [&].

File-tailoring statements

These are the lines that guide the file-tailoring services through the print utility skeletons, according to the type of print request.

)SEL *condition* paired with)ENDSEL	Select lines to be processed if the specified condition is true. Pairs of)SEL –)ENDSEL statements can be nested.
)SET	Set a variable to a specified value.
)DOT *tablename* paired with)ENDDOT	Include table elements to be processed from table *tablename*.
)CM *comments*	Comment lines.

Note: For details refer to *SDF II Run-Time Services*.

Print utility records

The print utility records in the print utility skeletons are basically structured like the logical lines in the print input file. However, print utility records contain the names of variables and table elements, whereas the logical lines already contain their actual contents.

Note: For the structure of the print input file and the relationship of the print utility records to the lines of the printed output, see “The print input file” on page 37.

The following is a description of the print utility records used to create the logical lines of the print input file:

General control record: The general control record specifies the print parameters. This record is required. It must be the first record in the print input file.

G *ppp width depth d*

G
Identifies the record as a general control record.

ppp
Placeholder for the page number. If the string specified here is found in the heading line, it is replaced with the actual page number.

width
Specifies the page width. The value must be at least 20. The first byte is used as the ASA control character.

depth
Specifies the physical page depth. The value must be at least 20.

d
Specifies the destination:

1 System printer
4 DBCS printer
5 DBCS with outlining

Heading records: The heading line is initially set to blank. The heading records allow you to modify the heading line.

Logically, the heading line is divided into segments of 30 bytes each. Single segments can be modified with heading records, leaving the other segments of the heading line unchanged.

When a heading record is encountered, output of the previous logical page is forced.

More than one heading record can be used to model the heading line. They take effect in the sequence in which they appear.

A special character combination, defined in the general control record, may be inserted in the heading line for the page number. This character combination is replaced by the appropriate value before printing the page.

```
H   <......................(76)......................>
Hn  <.......(30).......>
Hnm <.......(30).......><.......(30).......>
H0
H-
```

H Clear the heading line, then insert the supplied 76 bytes into it.

H*n* Overlay the *n*th section of the heading line with the supplied 30 bytes, where *n* is in the range of 1 through 9.

H*nm* Overlay *n*th section of the heading line with the first 30 bytes supplied. Overlay the *m*th section of the heading line with the second 30 bytes supplied, where *n* and *m* are in the range of 1 through 9.

H0 Do not change the heading line. The contents of the heading line are not affected. Output pages continue to be numbered. The only effect is to signal the start of a new page.

H- Suppress output of the heading line. The contents of the heading line are not affected. Output pages continue to be numbered. Output of heading lines is resumed after the next heading line record.

Subheading records and footing records: More than one subheading line or footing line may be specified. If the logical page is split into more than one physical page, the subheading portion and the footing portion of the page are repeated on subsequent pages.

The subheading lines and footing lines are cleared when a heading record is encountered in the input.

```
Snn <..................(nn <= 76)..................>
Snn+<..................(nn <= 76)..................>
Fnn <..................(nn <= 76)..................>
Fnn+<..................(nn <= 76)..................>
```

S*nn* Begin a new subheading line, *nn* bytes long.
S*nn*+ Add *nn* bytes to the previous subheading line.
F*nn* Begin a new footing line, *nn* bytes long.
F*nn*+ Add *nn* bytes to the previous footing line.

Text line records: Text line records are used to specify plain text lines. Plain text lines contain the text to be printed unchanged. Plain text lines that exceed the page width are truncated.

```
 <.......................(79).....................>
*<.......................(79).....................>
+<.......................(79).....................>
Tnn <..................(nn <= 76)..................>
Tnn+<..................(nn <= 76)..................>
```

blank Begin a new text line with a length of 79 bytes.

* Begin a new text line with a length of 79 bytes. This is useful for creating blank lines on the listing. Blank lines would otherwise be suppressed by file tailoring.

+ Add 79 bytes to the previous text line.

T*nn* Begin a new text line with a length of *nn* bytes.

T*nn*+ Add *nn* bytes to the previous text line.

Attribute records: Attribute records may follow any kind of record, except panel records. They specify byte-for-byte the attributes for the corresponding positions of the previous record.

```
Ann <..................(nn <= 76)..................>
Ann+<..................(nn <= 76)..................>
```

A*nn* Begin a new attribute line, *nn* bytes long.

A*nn*+ Add *nn* bytes to the previous attribute line.

There is one attribute byte for each text byte. The bits of an attribute byte have the following meanings, where 1 is the most significant bit and 8 is the least significant bit:

Bit	Meaning
1	Unused
2 – 3	Intensity: 0 Dark 1 Normal 2 Bright
4 – 5	Highlighting: 0 Normal 1 Blink 2 Reverse 3 Underscore
6 – 8	Color: 0 Default 1 Blue 2 Red 3 Pink 4 Green 5 Turquoise 6 Yellow 7 White

Panel records: Panel records describe a panel layout. Panel lines that exceed the page width are split.

Panels also contain attributes, which, depending on the abilities of the printer, may be reflected in the output.

```
PB width depth
PL<......................(78).....................>
PA<......................(78).....................>
PR<......................(78).....................>
```

PB *width depth* Start of panel information, where *width* and *depth* specify the dimension of the panel.

PL Panel line text for the complete panel. Each record is 78 bytes long. These records, which are required for panels, must follow the start of panel information record. They must describe the complete panel.

PA

Attribute information for the complete panel. Each record is 78 bytes long. These records, which are optional, must follow the panel line text records. There is one attribute byte for each text byte. The bits of an attribute byte have the following meanings, where 1 is the most significant bit and 8 is the least significant bit:

Bit	Meaning
1	Unused
2 – 3	Intensity: 0 Dark 1 Normal 2 Bright
4 – 5	Highlighting: 0 Normal 1 Blink 2 Reverse 3 Underscore
6 – 8	Color: 0 Default 1 Blue 2 Red 3 Pink 4 Green 5 Turquoise 6 Yellow 7 White

PR

Field ruling information for the complete panel. Each record is 78 bytes long. These records, which are optional, must follow the panel line text records or the panel attribute records. There is one attribute byte for each field ruling byte. The bits of a field ruling byte have the following meanings, where 1 is the most significant bit and 8 is the least significant bit:

Bit	Meaning
1	Character set (also set by the program for SO/SI): 0 EBCDIC 1 DBCS
2	DBCS character byte: 0 First byte 1 Second byte
3	Field ruling begin (set by the program)
4	Field ruling end (set by the program)
5	Field ruling left
6	Field ruling over
7	Field ruling right
8	Field ruling under

SDF2/DM variables and table elements

SDF2/DM variables and table elements are replaced by their assigned values during the first pass of the print utility.

Figure 9 lists the names of the SDF2/DM dialog variables and tables, their meanings, and the types of output for which they are used. In the skeletons, the names of variables and table elements are prefixed by an ampersand (&).

Figure 9 (Part 1 of 2). SDF2/DM dialog variables and tables, and their meanings

For all types of output	
IUV14OSY	Operating system (VSE)
IUV14FMT	Print format: **1** System printer **4** DBCS printer **5** DBCS printer, with outlining
Variables available except for DGIUS1IG	
IUV14PPN	Name of the object to be printed
IUV14PTY	Object type: **G** Panel group **P** Panel **S** Partition set **A** AID table **B** Control table
IUV14PLB	Object library
IUV14TSP	Object target system: **C** CICS/BMS **M** IMS/MFS **I** ISPF **G** GDDM–IMD **X** CSP/AD or VisualGen ***** All
IUV14DES	Description
IUV14DAT	Modification date
IUV14TIM	Modification time
IUV14USR	User who last modified
IUV14CMA	Used in some skeletons to select different parts
For output of panels	
IUV14PFN	Format name for the target system: MFS or All
IUV14ANM	The name of the scrollable area for the target system: ISPF or All
IUV14PDV	Device type
IUV14PPP	Physical page for the target system: MFS or All
For output of formats and layouts (DGIUS1P1, DGIUS1G5, DGIUS1S3)	
IUV10PDE	Panel depth

Figure 9 (Part 2 of 2). SDF2/DM dialog variables and tables, and their meanings

IUV10PWI	Panel width
IUI10PAN	SDF2/DM table containing the panel lines and attribute lines
IUI10PLI	SDF2/DM table element containing the panel lines
IUI10PA1	SDF2/DM table element containing the attributes for lines
IUI10PA2	SDF2/DM table element containing the ruling attributes for lines
For non-tabular output	
Use dialog variables. These dialog variables have the same names as the dialog variables in the associated panels. The names of the associated panels are given in the headers of the skeletons.	
For tabular output	
Use SDF2/DM tables with appropriate table elements. Table IUI10TXT with table elements IUI10T*nn* (*nn*=01, 02, ...) holds the contents of the columns. The maximum length of these table elements is 32 bytes. Skeleton DGIUS1W1 uses the prototype table directly. The names of the columns are given in the header of the skeleton.	

The print input file

The records of the print input file describe the lines of the printed output. More than one record of the print input file can be used to describe one line of the printed output. For example, the information of three heading records can be concatenated to form one heading line of the printed output.

The print input file consists of fixed-length records, 80 bytes long.

Figure 10 shows the records of the print input file.

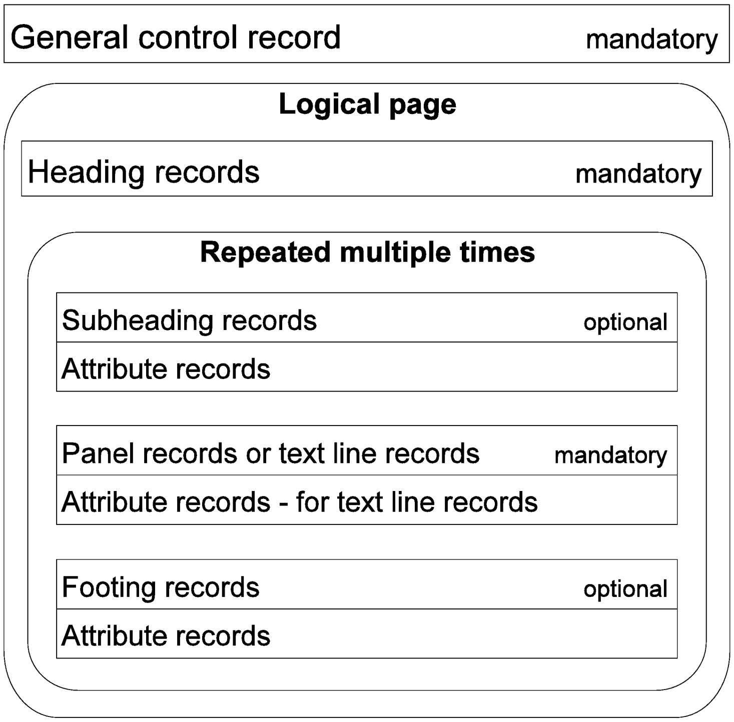


Figure 10. The structure of the print input file

The width of a logical page may be specified in the general control record at the beginning of the print input file. Records exceeding the defined width are truncated. The logical page depth is not restricted.

For panel output, records that exceed the page width are split into multiple pages, with the excess continued below.

A logical page of the print input file has the following records:

- The **heading record**, which is required, is the first record of a logical page. It triggers the beginning of a new logical page. It may contain the placeholder for the page number.
- **Subheading records**, which are optional, remain effective until the next block of subheading records. More than one subheading record may exist.
- **Panel records** or **text line records**:

  Text line records Contain the text to be printed unchanged. Text line records that exceed the page width are truncated.

  Panel records Describe formats. Panel line records that exceed the page width are split and are continued on the next line.

  If neither panel records nor text line records exist, the output of the page is suppressed.
- **Footing records**, which are optional, take effect from the end of the previous block of subheading records. More than one footing record may exist.
- **Attribute records** may follow any record, except a panel record. They immediately follow the record they describe.

Subheading records, text line records, and footing records may be intermixed freely. Each record of a print input file may contain DBCS strings enclosed in SO and SI characters. They must not wrap lines except for panels, where wrapping is allowed.

In the second pass of the SDF II print utility, the lines of the printed output are built according to the specifications in the print input file.

Making changed print utility skeletons available

Save changed skeletons in your customization library. How to give other users access to these skeletons is explained in “Making changes to keylists and skeletons available” on page 39.

Changing the SDF2/DM keylists

In SDF II, PF key assignments are held in SDF2/DM keylist tables in the SDF II installation library.

The naming convention for keylists is:

DGIYY*xx*.A

For more information about how to identify and modify keylists, refer to *SDF II Run-Time Services*.

Attention: Take care when you change keylists: they may be shared by several panels.

Making changed SDF2/DM keylist tables available

Save changed keylist tables in your customization library. How to give other users access to these skeletons is explained in "Making changes to keylists and skeletons available."

Making changes available to other users

When you have finished customizing SDF II, make changes available to other users, as described in this section.

Making changes to the user profile available

For each interactive user, SDF II maintains a user profile with the name *user*.PROFDGI1 in the library pointed to by the // LIBDEF PHASE,CATALOG statement in the DGIPROC.PROC procedure.

After setting your profile to the installation defaults, exit from SDF II, to refresh your profile in the library, then save the library member as the default profile for your installation.

To create a profile with your installation defaults for a new user, or to reset any changes made to an existing user profile, copy the installation default profile to *userid*.PROFDGI1.

Attention: Individual changes a user has made are lost.

Making changes to SDF2/DM tables available

Other customizable information is held in SDF2/DM tables that are common to all SDF II users. Customized tables are stored as members in the sublibrary you specify when you save your changes.

To make a customized table available for your installation, ensure that its library precedes the SDF II installation library in the LIBDEF search chain of the DGIPROC.PROC procedure.

After applying maintenance, you may need to customize the tables again.

Making changes to keylists and skeletons available

Save modified keylists or skeletons in your customization library.

To make a customized table available for your installation, ensure that the library precedes the SDF II installation library in the LIBDEF search chain of the DGIPROC.PROC procedure.

Chapter 5. Interfacing with SDF II

This chapter describes how to:

- Use the SDF II invocation interface
- Prepare a request member
- Use predefined panel elements
- Write a generation exit routine
- Write a data extraction and modification exit routine

SDF2/DM services are used to communicate between SDF II and the start-up and user exit REXX EXECs. For information about how to use SDF2/DM services, refer to *SDF II Run-Time Services*.

Using the invocation interface

SDF II provides an invocation interface, which can be used to invoke selected functions from a REXX EXEC. When you invoke a function, you describe the request to be performed by SDF II. On return, SDF II provides completion information.

By default, the SDF II batch jobs use the IBM-supplied start-up EXEC DGIIX. This EXEC requires a request file with the invocation parameters.

Preparing a request file for DGIIX

Put your SDF II statements in a library member or add it as SYSIPT data to the job. In the request member, include each request and its parameters. You can put more than one request in a request member. For each request, identify the object involved and any additional parameters required for the request to be successfully processed. If an error occurs, a message is written to SYSLST, and the request is terminated.

Each line of the request file has the form:

varname **=** **'***value***'**

where:

varname is the name of a variable.
value is the value to be assigned to *varname*.

Note: For check boxes, specify **/** to select the choice or a null string to deselect it.

When you include more than one request in a request file, indicate the start of the next request by entering an asterisk (*) in column 1, and leave the rest of the line blank. You can find examples of request files in “Invocation interface request examples” on page 41.

Invocation interface request examples

With the assignment statements in the request file, you assign values to SDF2/DM variables. You can assign to the SDF2/DM variables any value that you can enter in SDF II panel input fields.

For example, you can set a library identifier ILV10L0*x* in the request file. (*x* is in the range 1–9.) Thus the statement:

```
ILV10L02='PROJECT.GROUP'
```

will let you use the new library PROJECT.GROUP by setting the variable IIVLIB to 2.

Set the profile variables as described under “Specifying global variables for batch execution” on page 55.

Note: The figures in the examples in this section show the variable names for the panel input fields, not the input fields themselves.

Example 1: Generate a PL/I data structure and CICS/BMS macros

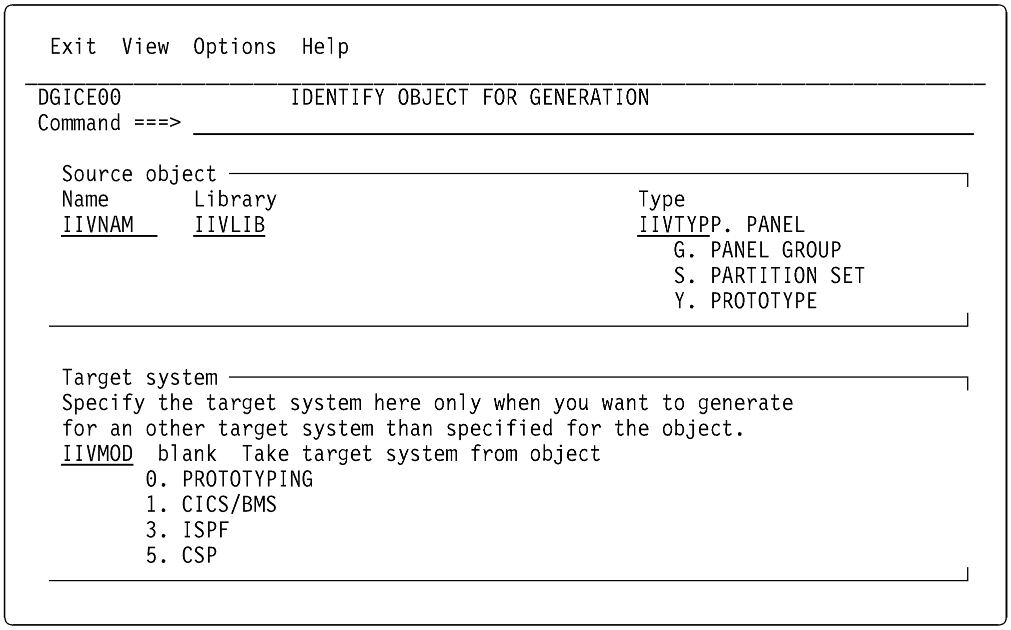
Figure 11 shows a request file for generating a PL/I data structure and CICS/BMS macros for the panel MYPAN. The generated output is to be stored in APPL1.DSECT and APPL1.MACRO.

```
IIVREQ   = 'G'           /* requests generation                         */
ILV10L01 = 'SDFOBJ.USER'  /* identifies your private library             */
ILV10L02 = 'SDFOBJ.GROUP' /* identifies your project library             */
IIVNAM   = 'MYPAN'       /* identifies the object name                  */
IIVLIB   = '*'           /* uses search order                           */
IIVTYP   = 'P'           /* identifies object as panel                  */
IIVMOD   = '1'           /* generates for CICS/BMS                      */
ICV10GDS = '/'           /* generates a data structure                  */
ICV10GCB = '/'           /* generates a control block source            */
ICV10DLA = 'PLI'         /* identifies programming language             */
ICV10DAL = ''            /* generates unaligned data structure          */
ICV10DGR = ''            /* generates no GRAPHIC declarations           */
ICV10DTR = ''            /* generates no TRIGRAPH declarations          */
ICV10ADI = ''            /* generates no SDF/CICS compatible adjuncts   */
ICV10CLG = ''            /* generates no length statement               */
ICV10DUE = ''            /* invokes no user exit routine                */
ICV10DVL = 'APPL1.DSECT'  /* identifies sublibrary for generated data str. */
ICV10DVT = 'A'           /* identifies the member type                  */
ICV10CDT = '*'           /* generates control block source for all devices */
ICV10CUE = ''            /* invokes no user exit routine                */
ICV10CVL = 'APPL1.MACRO'  /* identifies for control block source         */
ICV10CVT = 'A'           /* identifies the member type                  */
```

Figure 11. Request file to generate a PL/I data structure and CICS/BMS macros

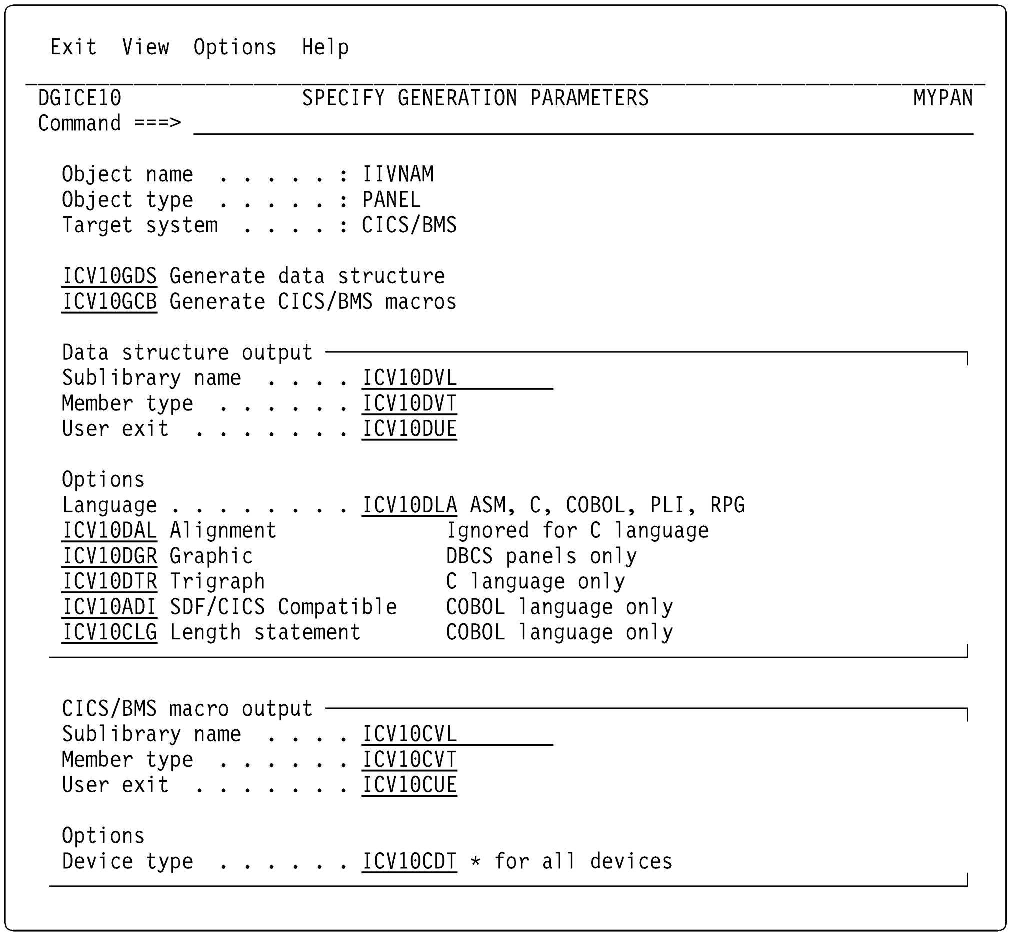
Figure 12 and Figure 13 on page 43 show the panels and the names of the dialog variables that you have set with the request file in this example.

The names of the variables shown in Figure 12 on page 43 for example 1 are the same for all requests for generation. This figure applies to examples 1 through 4 and will, therefore, not be repeated for examples 2 through 4.



```
 Exit  View  Options  Help
______________________________________________________________________________
DGICE00              IDENTIFY OBJECT FOR GENERATION
Command ===> _________________________________________________________________

  Source object ──────────────────────────────────────────────────────────────┐
  Name       Library                              Type
  IIVNAM     IIVLIB                               IIVTYPP. PANEL
                                                     G. PANEL GROUP
                                                     S. PARTITION SET
                                                     Y. PROTOTYPE
 ─────────────────────────────────────────────────────────────────────────────┘

  Target system ──────────────────────────────────────────────────────────────┐
  Specify the target system here only when you want to generate
  for an other target system than specified for the object.
  IIVMOD  blank  Take target system from object
         0. PROTOTYPING
         1. CICS/BMS
         3. ISPF
         5. CSP
 ─────────────────────────────────────────────────────────────────────────────┘
```

Figure 12. Variables for identifying an object for generation



```
 Exit  View  Options  Help
______________________________________________________________________________
DGICE10               SPECIFY GENERATION PARAMETERS                      MYPAN
Command ===> _________________________________________________________________

  Object name  . . . . . : IIVNAM
  Object type  . . . . . : PANEL
  Target system  . . . . : CICS/BMS

  ICV10GDS Generate data structure
  ICV10GCB Generate CICS/BMS macros

  Data structure output ──────────────────────────────────────────────────────┐
  Sublibrary name  . . . . ICV10DVL
  Member type  . . . . . . ICV10DVT
  User exit  . . . . . . . ICV10DUE

  Options
  Language . . . . . . . . ICV10DLA ASM, C, COBOL, PLI, RPG
  ICV10DAL Alignment              Ignored for C language
  ICV10DGR Graphic                DBCS panels only
  ICV10DTR Trigraph               C language only
  ICV10ADI SDF/CICS Compatible    COBOL language only
  ICV10CLG Length statement       COBOL language only
 ─────────────────────────────────────────────────────────────────────────────┘

  CICS/BMS macro output ──────────────────────────────────────────────────────┐
  Sublibrary name  . . . . ICV10CVL
  Member type  . . . . . . ICV10CVT
  User exit  . . . . . . . ICV10CUE

  Options
  Device type  . . . . . . ICV10CDT * for all devices
 ─────────────────────────────────────────────────────────────────────────────┘
```

Figure 13. Variables for generation parameters for CICS/BMS

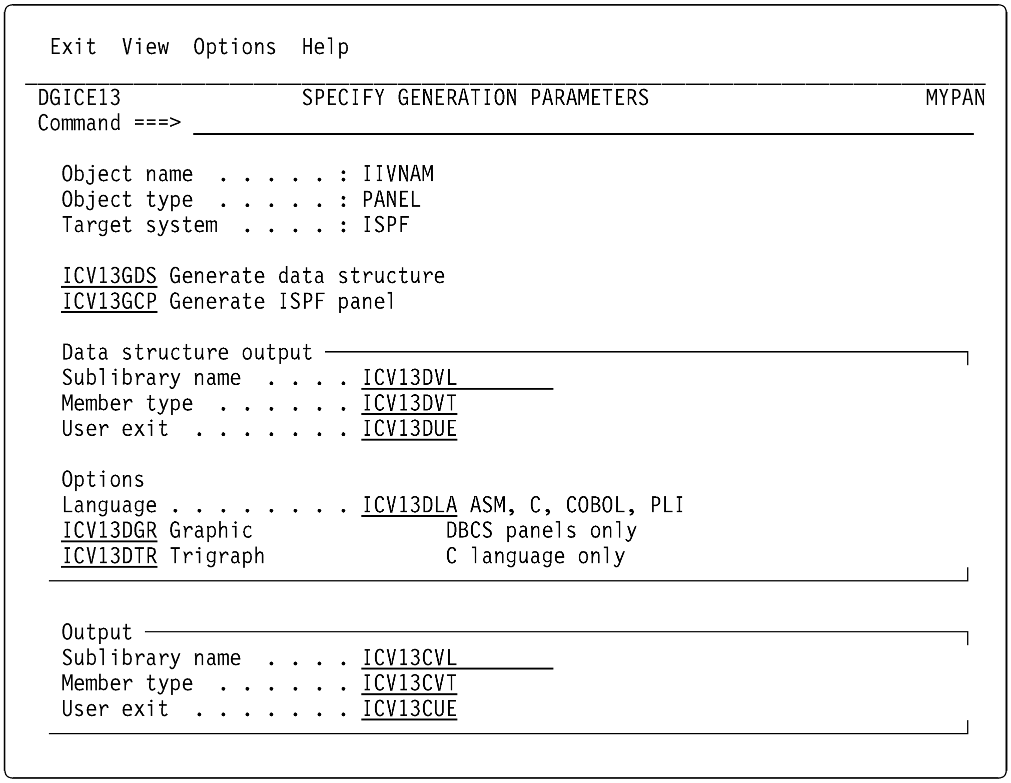
Example 2: Generate an ISPF panel

Figure 14 shows a request file for generating an ISPF panel for the panel MYPAN. The generated output is to be stored in APPL1.PANEL.

```
IIVREQ   = 'G'           /* requests generation                      */
ILV10L01 = 'SDFOBJ.USER'  /* identifies your private library          */
ILV10L02 = 'SDFOBJ.GROUP' /* identifies your project library          */
IIVNAM   = 'MYPAN'       /* identifies the object name               */
IIVLIB   = '*'           /* uses search order                        */
IIVTYP   = 'P'           /* identifies object as a panel             */
IIVMOD   = '3'           /* generates for ISPF                       */
ICV13GDS = '/'           /* generates data structure                 */
ICV13GCB = '/'           /* generates panel                          */
ICV13DLA = 'C'           /* identifies programming language          */
ICV13DGR = ''            /* generates no GRAPHIC declarations        */
ICV13DTR = ''            /* generates no TRIGRAPH declarations       */
ICV13DUE = ''            /* invokes no user exit routine             */
ICV13DVL = 'APPL1.DSECT'  /* identifies sublibrary for data structure */
ICV13DVT = 'A'           /* identifies the member type               */
ICV13CUE = ''            /* invokes no user exit routine             */
ICV13CVL = 'APPL1.PANEL'  /* identifies sublibrary for generated panel */
ICV13CVT = 'A'           /* identifies the member type               */
```

Figure 14. Request file to generate an ISPF panel

Figure 15 shows the panel and the names of the SDF2/DM variables that you have set with the request file in this example.



```
  Exit  View  Options  Help
 ____________________________________________________________________________
 DGICE13               SPECIFY GENERATION PARAMETERS                    MYPAN
 Command ===> _________________________________________________________________

   Object name  . . . . . : IIVNAM
   Object type  . . . . . : PANEL
   Target system  . . . . : ISPF

   ICV13GDS Generate data structure
   ICV13GCP Generate ISPF panel

   Data structure output ─────────────────────────────────────────────┐
   Sublibrary name  . . . . ICV13DVL________
   Member type  . . . . . . ICV13DVT
   User exit  . . . . . . . ICV13DUE

   Options
   Language . . . . . . . . ICV13DLA ASM, C, COBOL, PLI
   ICV13DGR Graphic              DBCS panels only
   ICV13DTR Trigraph             C language only
  ────────────────────────────────────────────────────────────────────┘

   Output ────────────────────────────────────────────────────────────┐
   Sublibrary name  . . . . ICV13CVL________
   Member type  . . . . . . ICV13CVT
   User exit  . . . . . . . ICV13CUE
  ────────────────────────────────────────────────────────────────────┘
```

Figure 15. Variables for generation parameters for ISPF

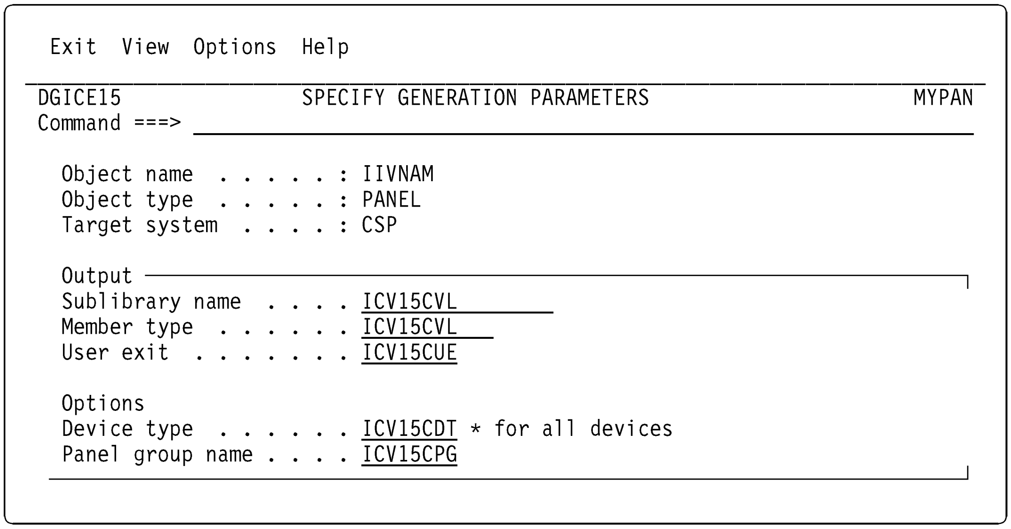
Example 3: Generate CSP/AD external source format

Figure 16 shows a request file for generating external source format for the panel MYPAN. The generated output is to be stored in APPL1.ESF.

```
IIVREQ   = 'G'            /* requests generation                    */
ILV10L01 = 'SDFOBJ.USER'  /* identifies your private library        */
ILV10L02 = 'SDFOBJ.GROUP' /* identifies your project library        */
IIVNAM   = 'MYPAN'        /* identifies the object name             */
IIVLIB   = '*'            /* uses search order                      */
IIVTYP   = 'P'            /* identifies the object as a panel       */
IIVMOD   = '5'            /* generates for CSP                      */
ICV15CVL = 'APPL1.ESF'    /* identifies sublibrary for generated panel */
ICV15CVT = 'A'            /* identifies the member type             */
ICV15CUE = ''             /* invokes no user exit routine           */
```

Figure 16. Request file to generate a CSP external source format

Figure 17 shows the panel and the names of the SDF2/DM variables that you have set with the request file in this example.



```
  Exit  View  Options  Help
 ______________________________________________________________________________
 DGICE15               SPECIFY GENERATION PARAMETERS                      MYPAN
 Command ===> _________________________________________________________________

   Object name  . . . . . : IIVNAM
   Object type  . . . . . : PANEL
   Target system  . . . . : CSP

   Output
   Sublibrary name  . . . . ICV15CVL
   Member type  . . . . . . ICV15CVL
   User exit  . . . . . . . ICV15CUE

   Options
   Device type  . . . . . . ICV15CDT * for all devices
   Panel group name . . . . ICV15CPG
```

Figure 17. Variables for generation parameters for CSP/AD

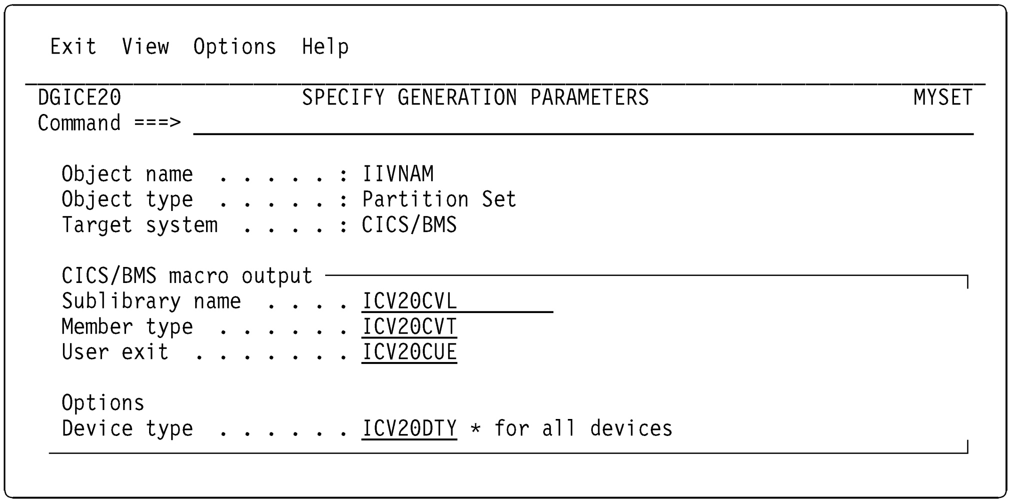
Example 4: Generate CICS/BMS partition set macros

Figure 18 shows a request file for generating the partition set MYPSET. The generated output is to be stored in APPL1.MACRO.

```
IIVREQ   = 'G'           /* requests generation                           */
ILV10L01 = 'SDFOBJ.USER'  /* identifies your private library              */
ILV10L02 = 'SDFOBJ.GROUP' /* identifies your project library              */
IIVNAM   = 'MYPSET'      /* identifies the object name                    */
IIVLIB   = '*'           /* uses search order                             */
IIVTYP   = 'S'           /* identifies object as a partition set          */
IIVMOD   = '1'           /* generates for CICS/BMS                        */
ICV20DTY = '*'           /* generates control block source for all devices */
ICV20CUE = ''            /* invokes no user exit routine                  */
ICV20CVL = 'APPL1.MACRO'  /* identifies sublib. for generated partition set */
ICV20CVT = 'A'           /* identifies the member type                    */
```

Figure 18. Request file to generate a BMS partition set

Figure 19 shows the panel and the names of the SDF2/DM variables that you have set with the request file in this example.



```
  Exit  View  Options  Help
 ________________________________________________________________________________
 DGICE20              SPECIFY GENERATION PARAMETERS                        MYSET
 Command ===> __________________________________________________________________

   Object name  . . . . . : IIVNAM
   Object type  . . . . . : Partition Set
   Target system  . . . . : CICS/BMS

   CICS/BMS macro output
   Sublibrary name  . . . . ICV20CVL________
   Member type  . . . . . . ICV20CVT
   User exit  . . . . . . . ICV20CUE

   Options
   Device type  . . . . . . ICV20DTY * for all devices
```

Figure 19. Variables for generation parameters for CICS/BMS

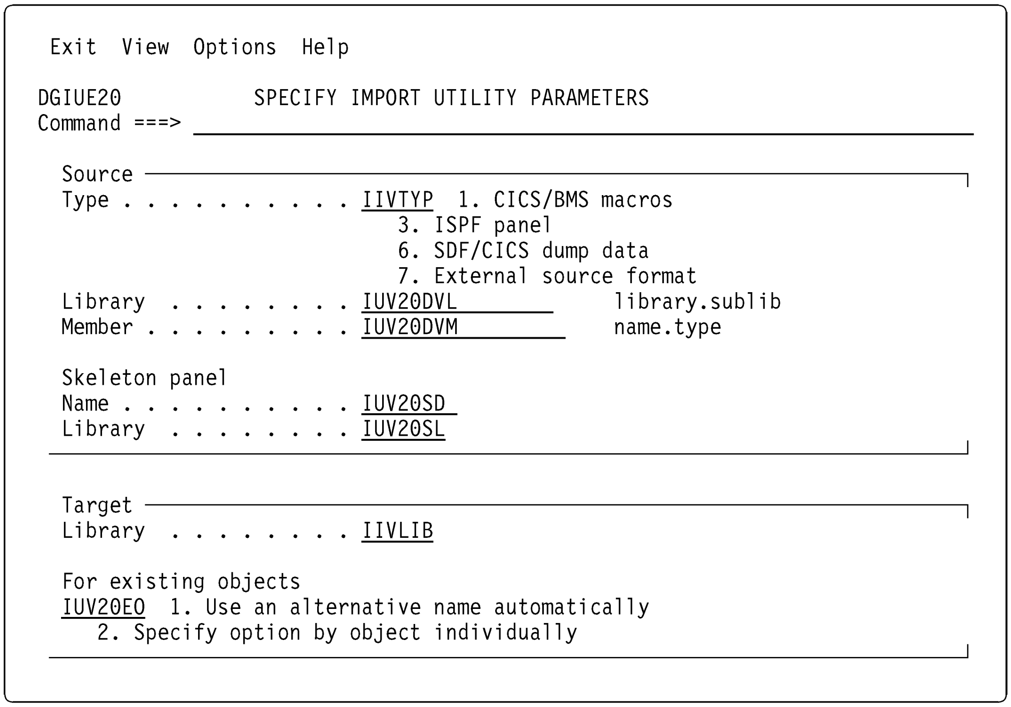
Example 5: Migrate SDF/CICS dump data to SDF II

Figure 20 shows a request file for migrating an SDF/CICS dump data.

```
ILV10L01 = 'SDFOBJ.USER'           /* library 1                  */
IIVREQ='M'                         /* request=import             */
IIVTYP='6'                         /* type=SDF/CICS object       */
IUV20DVL='SDFOBJ.MIGRATE'          /* L.S of input member        */
IUV20DVM='MIGRATE.DATA'            /* M.T of input member        */
IIVLIB='1'                         /* to library 1 specified above  */
```

Figure 20. Request file to import SDF/CICS dump data

Figure 21 shows the panel and the names of the SDF2/DM variables that you have set with the request file in this example.



```
  Exit  View  Options  Help

 DGIUE20          SPECIFY IMPORT UTILITY PARAMETERS
 Command ===> ________________________________________________________________

   Source ───────────────────────────────────────────────────────────────────┐
   Type . . . . . . . . . . IIVTYP  1. CICS/BMS macros
                               3. ISPF panel
                               6. SDF/CICS dump data
                               7. External source format
   Library  . . . . . . . . IIVL20DVL             library.sublib
   Member . . . . . . . . . IUV20DVM              name.type

   Skeleton panel
   Name . . . . . . . . . . IUV20SD
   Library  . . . . . . . . IUV20SL
 ────────────────────────────────────────────────────────────────────────────┘

   Target ───────────────────────────────────────────────────────────────────┐
   Library  . . . . . . . . IIVLIB

   For existing objects
   IUV20EO  1. Use an alternative name automatically
      2. Specify option by object individually
 ────────────────────────────────────────────────────────────────────────────┘
```

Figure 21. Variables for import utility parameters for SDF/CICS

Note: If an object already exists, SDF II generates an alternative name automatically. You cannot influence this value. You will find the name in the message list.

For a sample job, see “Migrating objects from SDF/CICS” on page 85.

Example 6: Print an object

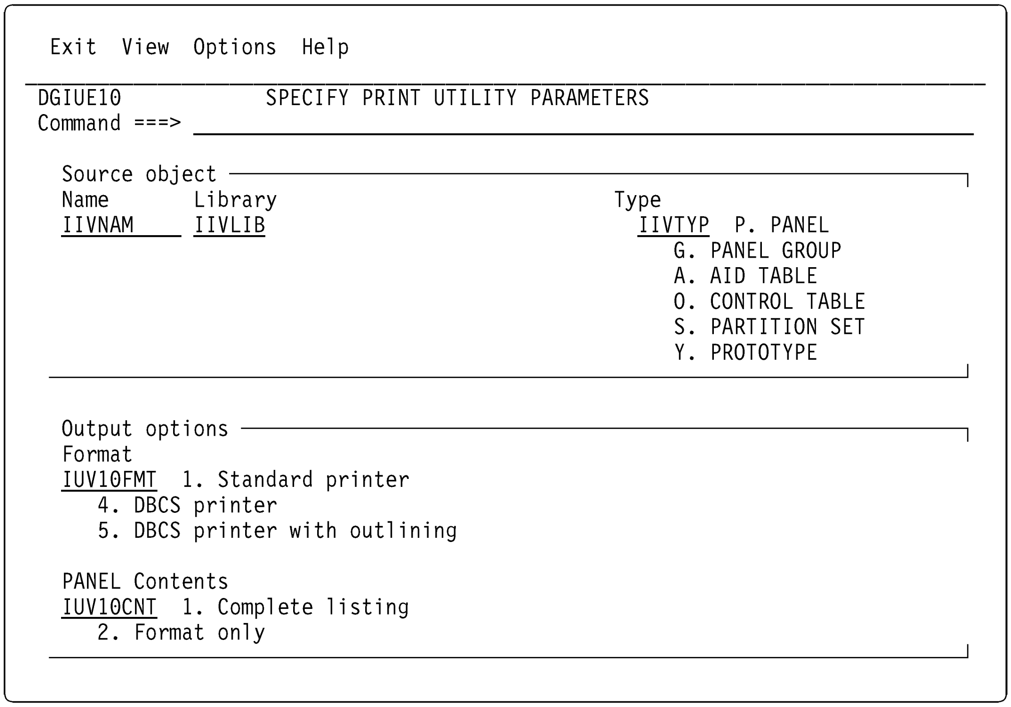
Figure 22 shows a request file for printing a complete listing of the panel MYPANEL.

To print other objects, specify the appropriate value for IIVTYP. The output is to be formatted for the standard printer.

```
IIVREQ   = 'P'            /* requests printing                  */
ILV10L01 = 'SDFOBJ.USER'  /* identifies your private library    */
ILV10L02 = 'SDFOBJ.GROUP' /* identifies your project library    */
IIVNAM   = 'MYPANEL'      /* identifies the object name         */
IIVLIB   = '*'            /* uses search order                  */
IIVTYP   = 'P'            /* identifies object as a panel       */
IUV10FMT = '1'            /* formats output for standard printer */
IUV10CNT = '1'            /* requests complete listing          */
```

Figure 22. Request file to print a panel

Figure 23 shows the panel and the names of the SDF2/DM variables that you have set with the request file in this example.



```
  Exit  View  Options  Help
 ________________________________________________________________________________
 DGIUE10           SPECIFY PRINT UTILITY PARAMETERS
 Command ===> ___________________________________________________________________

   Source object ──────────────────────────────────────────────────────────────┐
   Name       Library                             Type
   IIVNAM___  IIVLIB                                IIVTYP  P. PANEL
                                                       G. PANEL GROUP
                                                       A. AID TABLE
                                                       O. CONTROL TABLE
                                                       S. PARTITION SET
                                                       Y. PROTOTYPE
  ─────────────────────────────────────────────────────────────────────────────┘

   Output options ─────────────────────────────────────────────────────────────┐
   Format
   IUV10FMT  1. Standard printer
      4. DBCS printer
      5. DBCS printer with outlining

   PANEL Contents
   IUV10CNT  1. Complete listing
      2. Format only
  ─────────────────────────────────────────────────────────────────────────────┘
```

Figure 23. Variables for print utility parameters for a panel

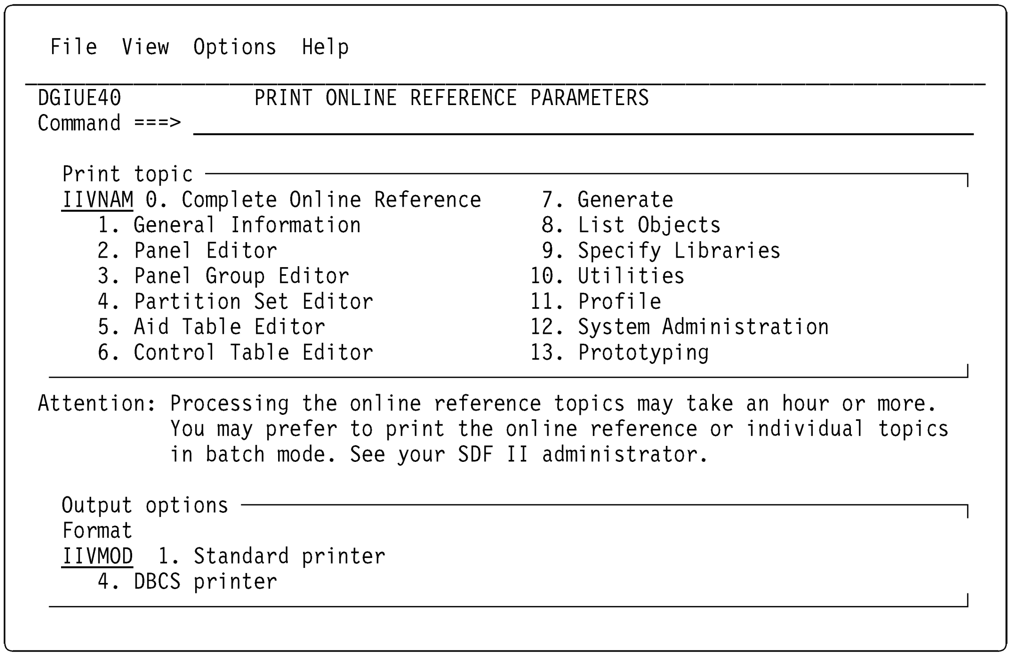
Example 7: Print a chapter of the online reference

Figure 24 shows a request file for printing the Panel Editor topic of the online reference. The output is to be prepared for the standard printer.

```
IIVREQ   = 'O'          /* requests printing online ref (use letter O)  */
IIVNAM   = '2'          /* prints the panel editor topic                */
IIVMOD   = '1'          /* prepares output for the standard printer     */
```

Figure 24. Request file to print a topic of the online reference

Figure 25 shows the panel and the names of the SDF2/DM variables that you have set with the request file in this example.



```
  File  View  Options  Help
 ________________________________________________________________________________
 DGIUE40          PRINT ONLINE REFERENCE PARAMETERS
 Command ===> _________________________________________________________________

   Print topic ──────────────────────────────────────────────────────────────┐
   IIVNAM 0. Complete Online Reference     7. Generate
      1. General Information               8. List Objects
      2. Panel Editor                      9. Specify Libraries
      3. Panel Group Editor               10. Utilities
      4. Partition Set Editor             11. Profile
      5. Aid Table Editor                 12. System Administration
      6. Control Table Editor             13. Prototyping
  ───────────────────────────────────────────────────────────────────────────┘
 Attention: Processing the online reference topics may take an hour or more.
            You may prefer to print the online reference or individual topics
            in batch mode. See your SDF II administrator.

   Output options ───────────────────────────────────────────────────────────┐
   Format
   IIVMOD  1. Standard printer
      4. DBCS printer
  ───────────────────────────────────────────────────────────────────────────┘
```

Figure 25. Variables for printing the online reference parameters

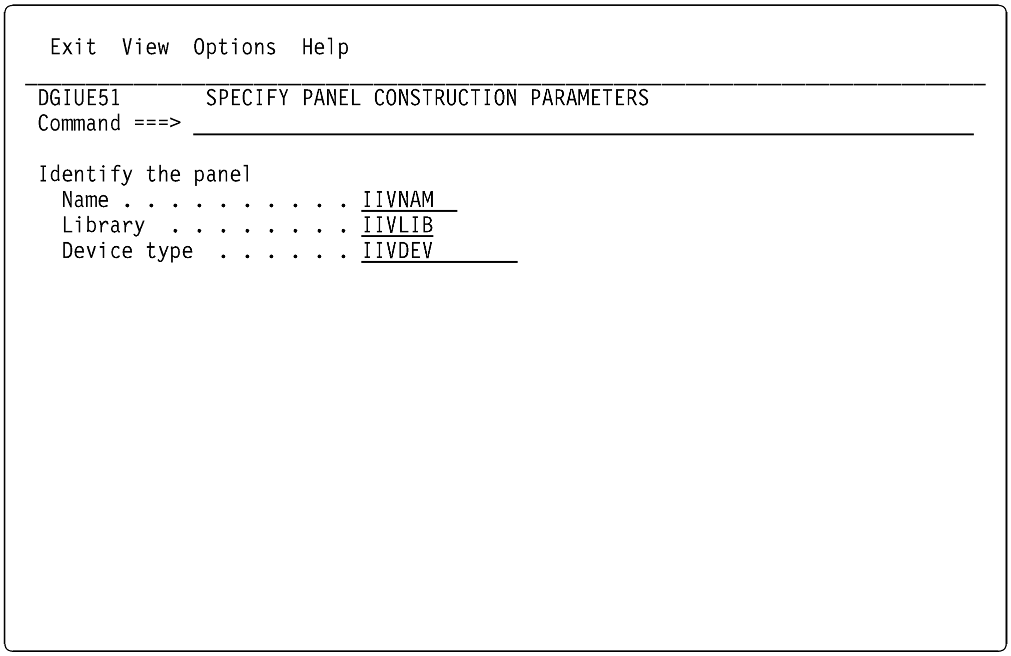
Example 8: Construct a panel

Figure 26 shows a request file for constructing a panel from predefined elements, such as fields, arrays, or repeat formats. You define these elements in the construction utility table, whose name (default DGIU5TAB) you specify in the field IIVONAM of the request file. The contents of IIVOLIB are explained in “Using the invocation interface” on page 40.

```
IIVREQ   = 'A'              /* requests construction              */
ILV10L01 = 'SDFOBJ.USER'    /* identifies your private library    */
IIVNAM   = 'MYPAN'          /* identifies the object name         */
IIVLIB   = '1'              /* stores panel in library 1          */
IIVDEV   = '3270'           /* identifies device type             */
IIVONAM  = 'DGIU5TAB'       /* identifies construction utility table */
IIVOLIB  = '0240800HELPPAN ' /* 024 = depth of panel format        */
                            /* 080 = width of panel format        */
                            /* 0 = include a command line         */
                            /* HELPPAN = name of help panel       */
```

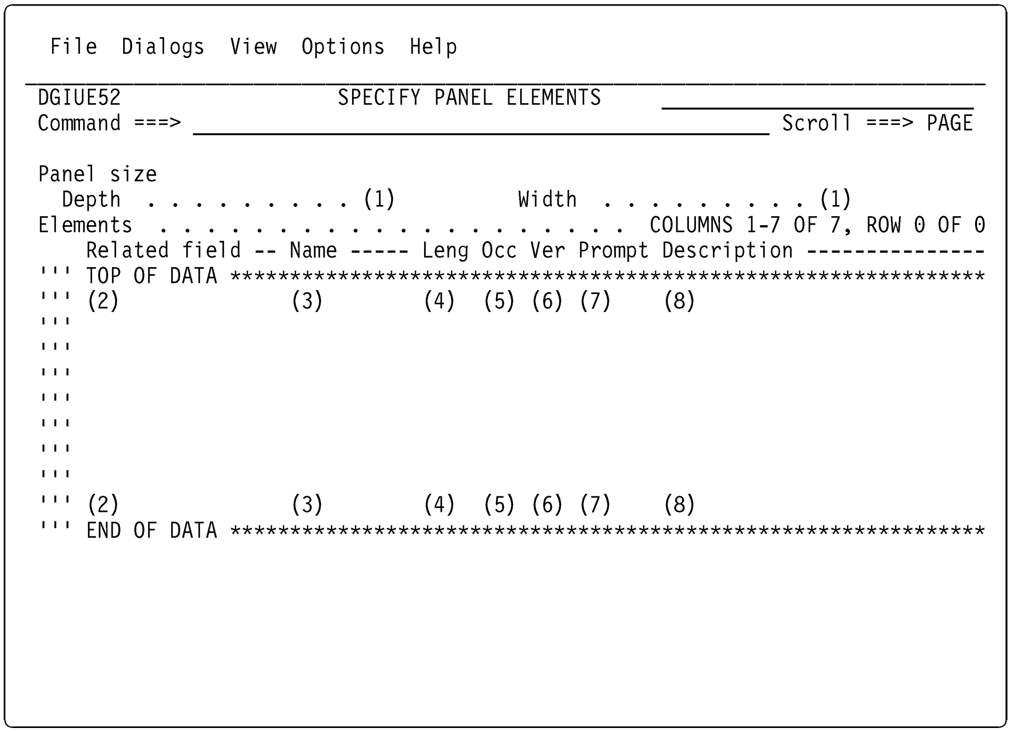
Figure 26. Request file to construct a panel

Figure 27 and Figure 28 on page 51 show the panels and the names of the SDF2/DM variables that you have set with the request file and the construction utility table in this example.



```
  Exit  View  Options  Help
 ________________________________________________________________________________
 DGIUE51       SPECIFY PANEL CONSTRUCTION PARAMETERS
 Command ===> __________________________________________________________________

 Identify the panel
   Name . . . . . . . . . . . IIVNAM__
   Library  . . . . . . . . . IIVLIB
   Device type  . . . . . . . IIVDEV_______
```

Figure 27. Variables for panel construction parameters



```
 File  Dialogs  View  Options  Help
______________________________________________________________________________
DGIUE52                 SPECIFY PANEL ELEMENTS     __________________________
Command ===> _______________________________________________ Scroll ===> PAGE

Panel size
  Depth  . . . . . . . . . (1)          Width  . . . . . . . . . (1)
Elements  . . . . . . . . . . . . . . . . . . . . COLUMNS 1-7 OF 7, ROW 0 OF 0
    Related field -- Name ----- Leng Occ Ver Prompt Description ---------------
''' TOP OF DATA ***************************************************************
''' (2)              (3)        (4)  (5) (6) (7)    (8)
'''
'''
'''
'''
'''
'''
'''
''' (2)              (3)        (4)  (5) (6) (7)    (8)
''' END OF DATA ***************************************************************
```

Figure 28. Variables for panel elements for panel construction

The numbers (*n*) in Figure 28 indicate the following variables:

(1)	IIVOLIB
(2)	IUV5RFD
(3)	IUV5NAM
(4)	IUV5LEN
(5)	IUV5OCC
(6)	IUV5VER
(7)	IUV5PRO
(8)	IUV5DES

These variables are set from your specifications in the corresponding columns of the invocation interface table (default name DGIIFTAB) and the construction utility table. See Figure 38 on page 60 and Figure 39 on page 65 for a description of these SDF2/DM tables.

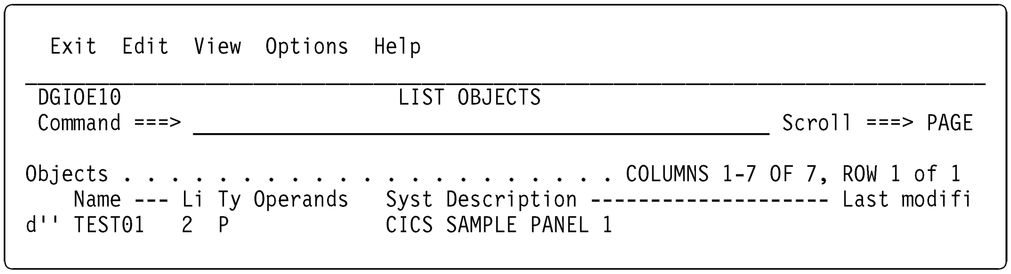
Example 9: Delete an object

Figure 29 shows a request file for deleting an object.

```
IIVREQ   = 'D'           /* requests deletion                   */
ILV10L02 = 'SDFOBJ.GROUP' /* identifies your project library     */
IIVNAM   = 'TEST01'      /* identifies the input object name    */
IIVLIB   = '2'           /* identifies the input library number */
IIVTYP   = 'P'           /* identifies object type as panel     */
```

Figure 29. Request file to delete an object

Figure 30 shows how the same operation would be done online; note that there are no corresponding variables.



```
   Exit  Edit  View  Options  Help
________________________________________________________________________________
 DGIOE10                        LIST OBJECTS
 Command ===> ________________________________________________ Scroll ===> PAGE

Objects . . . . . . . . . . . . . . . . . . . . . COLUMNS 1-7 OF 7, ROW 1 of 1
    Name --- Li Ty Operands   Syst Description -------------------- Last modifi
d'' TEST01   2  P             CICS SAMPLE PANEL 1
```

Figure 30. List Objects panel for deleting an object

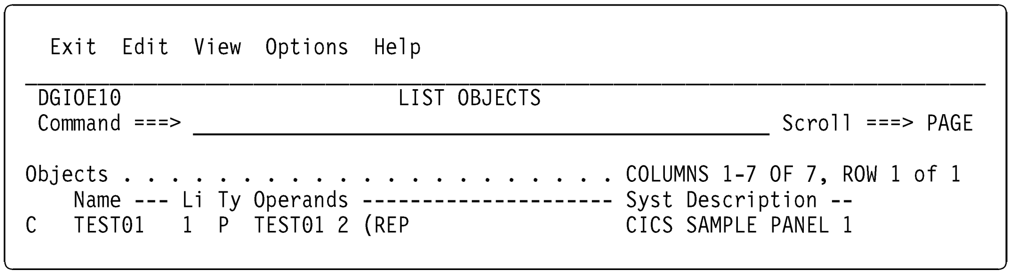
Example 10: Copy an object

Figure 31 shows a request file for copying a panel. IIVMOD specifies that an existing panel will be replaced.

```
IIVREQ   = 'C'           /* requests copy                          */
ILV10L01 = 'SDFOBJ.USER'  /* identifies your private library        */
ILV10L02 = 'SDFOBJ.GROUP' /* identifies your project library        */
IIVNAM   = 'TEST01'       /* identifies the input object name       */
IIVLIB   = '1'            /* identifies the input library number    */
IIVTYP   = 'P'            /* identifies the object type: panel      */
IIVONAM  = 'TEST01'       /* identifies the output object name      */
IIVOLIB  = '2'            /* identifies the output library number   */
IIVMOD   = '10'           /* requests replacement of existing object */
```

Figure 31. Request file to copy an object

Figure 32 shows how the same operation would be done online; note that there are no corresponding variables.



```
   Exit  Edit  View  Options  Help
 ______________________________________________________________________________
 DGIOE10                       LIST OBJECTS
 Command ===> ________________________________________________ Scroll ===> PAGE

Objects . . . . . . . . . . . . . . . . . . . . . . COLUMNS 1-7 OF 7, ROW 1 of 1
    Name --- Li Ty Operands --------------------- Syst Description --
C   TEST01   1  P  TEST01 2 (REP                  CICS SAMPLE PANEL 1
```

Figure 32. List Objects panel for copying an object

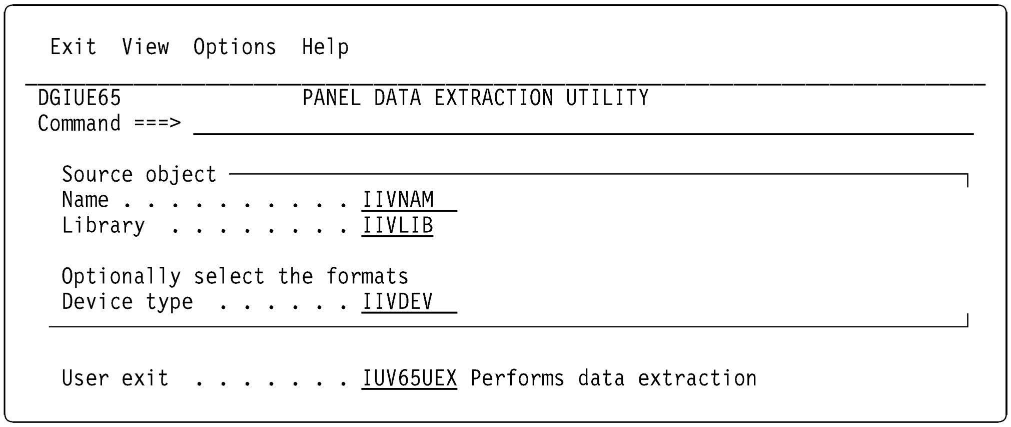
Example 11: Extract panel data

Figure 33 shows a request file for extracting panel data.

```
IIVREQ   = 'X'          /* requests extraction                */
ILV10L01 = 'SDFOBJ.USER' /* identifies your private library     */
IIVNAM   = 'TEST01'     /* identifies the input object name    */
IIVLIB   = '1'          /* identifies the input library number */
IIVDEV   = '3270'       /* identifies the device type          */
IIVTYP   = 'P'          /* identifies the object type: panel   */
IUV65UEX = 'DGIUX60C'   /* invokes a user exit routine         */
```

Figure 33. Request file to extract a panel data

Figure 34 shows the panel and the names of SDF2/DM variables that you have set with the request file in this example.



```
 Exit  View  Options  Help
______________________________________________________________________________
DGIUE65              PANEL DATA EXTRACTION UTILITY
Command ===> _________________________________________________________________

  Source object ──────────────────────────────────────────────────────────┐
  Name . . . . . . . . . . IIVNAM
  Library  . . . . . . . . IIVLIB

  Optionally select the formats
  Device type  . . . . . . IIVDEV
 ──────────────────────────────────────────────────────────────────────────┘

  User exit  . . . . . . . IUV65UEX Performs data extraction
```

Figure 34. Variables for extracting panel data

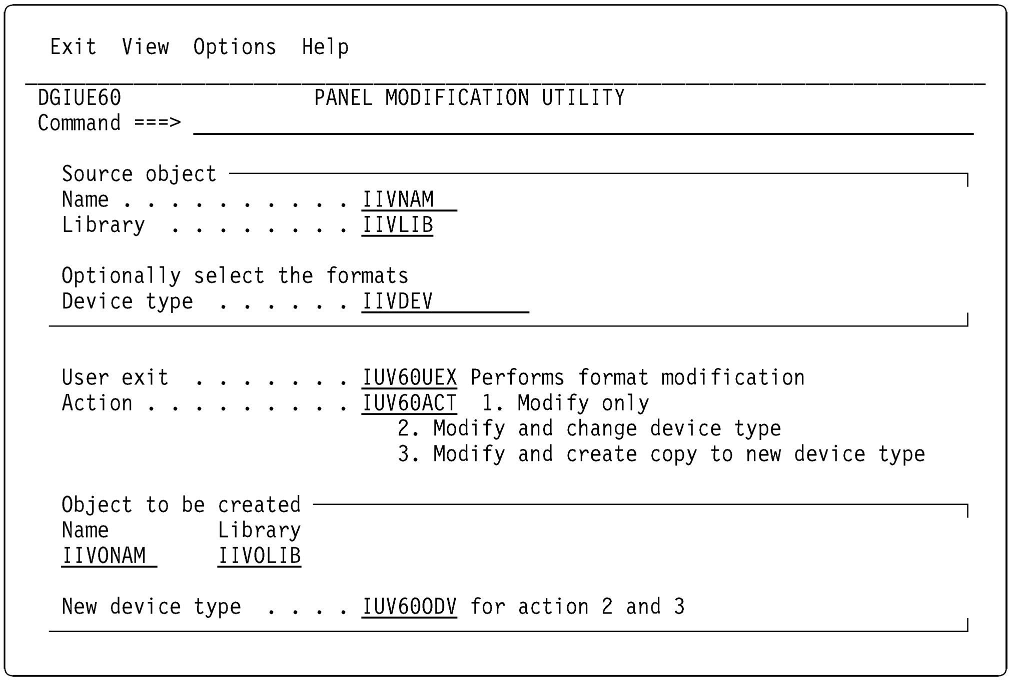
Example 12: Modify a panel

Figure 35 shows a request file for modifying a panel.

```
IIVREQ   = 'F'           /* requests modification              */
ILV10L01 = 'SDFOBJ.USER'  /* identifies your private library      */
ILV10L02 = 'SDFOBJ.GROUP' /* identifies your project library      */
IIVNAM   = 'TEST01'      /* identifies the input object name     */
IIVLIB   = '1'           /* identifies the input library number  */
IIVDEV   = '3270'        /* identifies the device type           */
IIVTYP   = 'P'           /* identifies the object type: panel    */
IIVONAM  = 'TEST01'      /* identifies the output object name    */
IIVOLIB  = '2'           /* identifies the output library number */
IUV60ACT = '1'           /* requests modify only                 */
IUV60UEX = 'DGIUX60C'    /* invokes a user exit routine          */
```

Figure 35. Request file to modify a panel

Figure 36 shows the panel and the names of SDF2/DM variables that you have set with the request file in this example.



```
  Exit  View  Options  Help
 ______________________________________________________________________________
 DGIUE60                PANEL MODIFICATION UTILITY
 Command ===> ______________________________________________________________

   Source object ─────────────────────────────────────────────────────────┐
   Name . . . . . . . . . . . IIVNAM
   Library  . . . . . . . . . IIVLIB

   Optionally select the formats
   Device type  . . . . . . . IIVDEV
  ────────────────────────────────────────────────────────────────────────┘

   User exit  . . . . . . . . IUV60UEX Performs format modification
   Action . . . . . . . . . . IUV60ACT  1. Modify only
                                 2. Modify and change device type
                                 3. Modify and create copy to new device type

   Object to be created ──────────────────────────────────────────────────┐
   Name         Library
   IIVONAM      IIVOLIB

   New device type  . . . . . IUV60DV for action 2 and 3
  ────────────────────────────────────────────────────────────────────────┘
```

Figure 36. Variables for modifying a panel

Specifying global variables for batch execution

Global variables can be used in all batch jobs. Any assignment remains active for the rest of the job, unless changed by another assignment.

ILV10L01 through ILV10L09 — Library name

Specify up to nine sublibraries to be used in subsequent requests. (The final digit of the variable names corresponds to a library ID on the Specify Libraries panel.) In a request, the variables IIVLIB and IIVOLIB can be set to a library ID to which a sublibrary has been assigned.

Specify the name of the sublibrary in the form:

library.sublib

ILV10D01 through ILV10D09 — Description

Optionally specify a short description of the library.

ILV10L10 — Autosave library

Specify the name of the library to be used for automatically saving objects. You can use one of the libraries specified in the list of object libraries. The default is to have no autosave library.

ILV10LRO — Search related objects

Specify which libraries are to be searched for related objects, such as included panels. Three choices are available:

1. Search all libraries

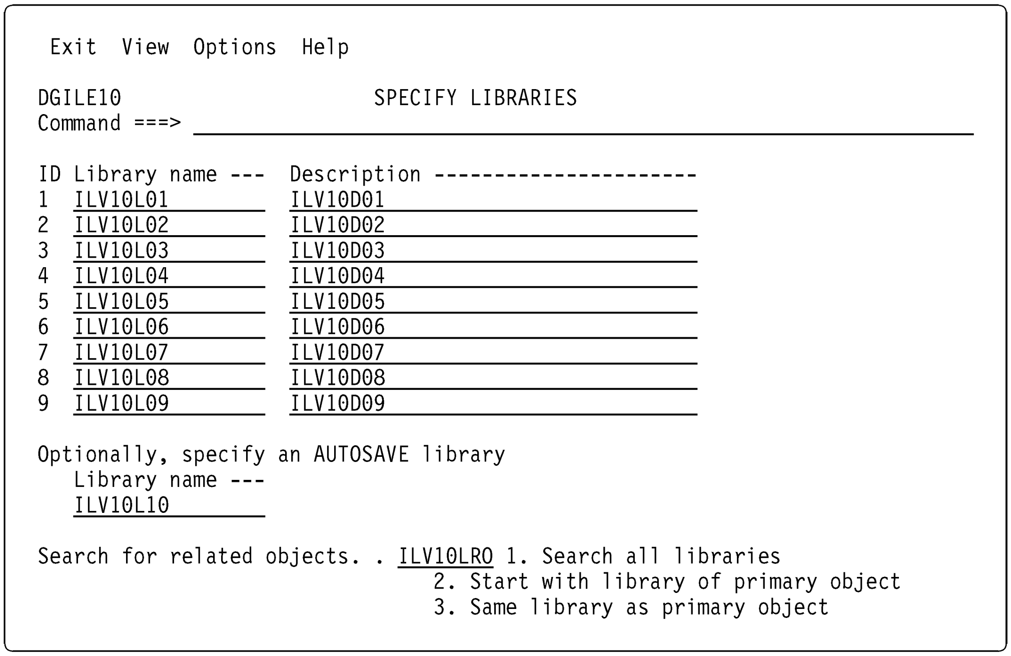
   Related objects can be in any of the libraries. This is the default.

2. Start with library of primary object

   Related objects can be in the same library as the object that refers to them and in any library that comes later in the search order. Library 9 is the last to be searched.

3. Same library as primary object

   Related objects must be in the same library as the object that refers to them.



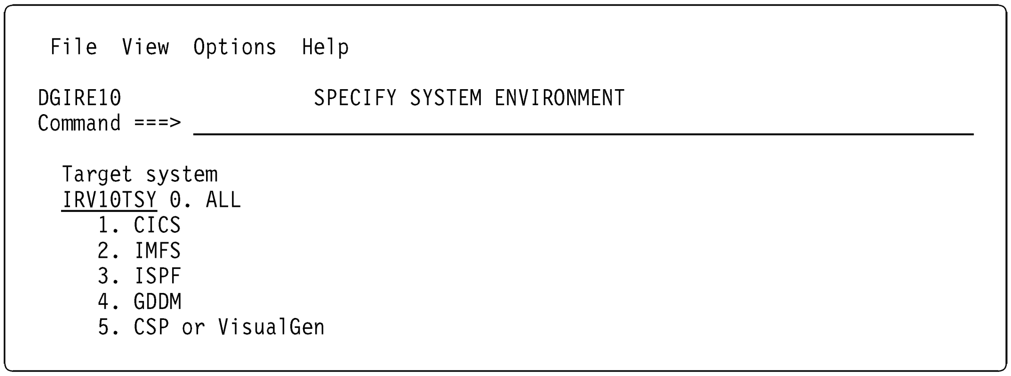
```
 Exit  View  Options  Help

DGILE10                    SPECIFY LIBRARIES
Command ===> _________________________________________________________________

ID Library name ---  Description ----------------------
1  ILV10L01________  ILV10D01__________________________
2  ILV10L02________  ILV10D02__________________________
3  ILV10L03________  ILV10D03__________________________
4  ILV10L04________  ILV10D04__________________________
5  ILV10L05________  ILV10D05__________________________
6  ILV10L06________  ILV10D06__________________________
7  ILV10L07________  ILV10D07__________________________
8  ILV10L08________  ILV10D08__________________________
9  ILV10L09________  ILV10D09__________________________

Optionally, specify an AUTOSAVE library
   Library name ---
   ILV10L10________

Search for related objects. . ILV10LRO 1. Search all libraries
                                 2. Start with library of primary object
                                 3. Same library as primary object
```

Target System Environment

IRV10TSY — Target system

Specify the target system for which you want to create a new SDF II object:

0	ALL
1	CICS
2	IMFS
3	ISPF
4	GDDM
5	CSP



```
 File  View  Options  Help

DGIRE10                SPECIFY SYSTEM ENVIRONMENT
Command ===> ____________________________________________________________________

  Target system
  IRV10TSY 0. ALL
     1. CICS
     2. IMFS
     3. ISPF
     4. GDDM
     5. CSP or VisualGen
```

Printer Defaults

In this dialog, you can define the size of the page used by the print utility.

Page depth

The print page length can be set according to the print page length available on the printer.

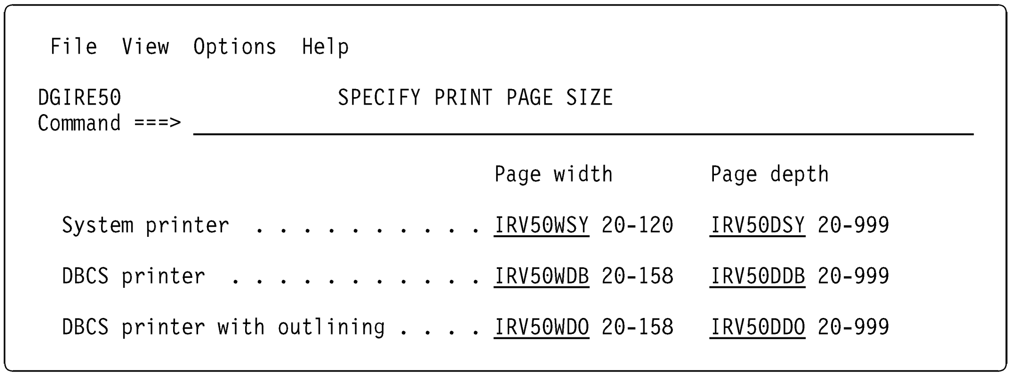
Page width

The print page width can be set according to the paper sizes available on the printer.

Printer	Range of widths
System printer	20 through 120
DBCS printer	20 through 158
DBCS printer with outlining	20 through 158

SDF II truncates any line that exceeds the width you specify. It also splits formats to fit the print width.

If you change the page width and want to print complete objects—not just the formats—adjust the skeletons defining the print layout.



```
 File  View  Options  Help

DGIRE50                  SPECIFY PRINT PAGE SIZE
Command ===> ____________________________________________________________________

                                      Page width        Page depth

  System printer  . . . . . . . . . . IRV50WSY 20-120   IRV50DSY 20-999

  DBCS printer  . . . . . . . . . . . IRV50WDB 20-158   IRV50DDB 20-999

  DBCS printer with outlining . . . . IRV50WDO 20-158   IRV50DDO 20-999
```

Writing a start-up REXX EXEC

This section describes the invocation interface table. It also explains how to write an invocation REXX EXEC, which you can use instead of the IBM-supplied DGIIX EXEC. This EXEC must use SDF2/DM services to pass data to SDF II.

A start-up REXX EXEC must define the invocation interface table and must invoke SDF II.

When you have prepared the SDF2/DM table (see “The invocation interface table” on page 60), call SDF II with:

DGIEXEC SELECT CMD(DGIIXINV SDF2I [*tablename*]**)**

where:

tablename is the name of the invocation interface SDF2/DM table. The default is DGIIFTAB.

Figure 37 on page 59 is an example of a REXX EXEC. It creates the table and invokes SDF II.

```
/* ------------------------ REXX ---------------------------------- */
/* Create the SDF2/DM table DGIIFTAB and invoke SDF II              */
/* ---------------------------------------------------------------- */
   table = 'DGIIFTAB'               /* Function table name          */
   names = ,                        /* Column names                 */
           'IIVROW',
           'IIVREQ',
           'IIVTYP',
           'IIVNAM',
           'IIVLIB',
           'IIVONAM',
           'IIVOLIB',
           'IIVDEV',
           'IIVMOD',
           'IIVPRO'
/* ---------------------------------------------------------------- */
/* Build SDF2/DM Table                                              */
/* ---------------------------------------------------------------- */
   address DGIEXEC                /* Address DGIEXEC interface      */
   "CONTROL ERRORS RETURN"        /* Return to REXX on errors       */
   "TBCREATE" table "NAMES("names") NOWRITE REPLACE" /* create table*/
   if rc>4 then signal error1     /* If create fails: error message */
/* ---------------------------------------------------------------- */
/* Fill Table Variables                                             */
/* ---------------------------------------------------------------- */
   iivrow  = "V"                  /* It is a request row            */
   iivreq  = "E"                  /* Edit an object                 */
   iivnam  = "MYPAN"              /* Object name is MYPAN           */
   iivtyp  = "P"                  /* It is a panel                  */
   iivlib  = "1"                  /* It is on library 1             */
   iivonam = "MYDEFP"             /* Use default panel MYDEFP       */
   iivolib = "1"                  /* Default panel is on library 1  */
   iivpro  = "G"                  /* Do not prompt for parameter    */
/* ---------------------------------------------------------------- */
/* Fill table, call SDF II and discard table                        */
/* ---------------------------------------------------------------- */
   "TBADD" table                  /* Add the row                    */
   if rc>0 then signal error3     /* If tbadd  fails: error message */
   "SELECT CMD(DGIIXINV SDF2I)"   /* Call SDF II                    */
   irc=rc                         /* Save retcode                   */
   "TBEND" table                  /* Discard table                  */
   exit irc                       /* Exit program                   */
/* ---------------------------------------------------------------- */
/* Error Exits                                                      */
/* ---------------------------------------------------------------- */
   error1:
           say 'Return Code' rc 'from TBCREATE' table
           exit 12                /* Exit program                   */
   error3:
           say 'Return Code' rc 'from TBADD' table
           exit 12                /* Exit program                   */
/* ---------------------------------------------------------------- */
```

Figure 37. REXX example for the SDF2 invocation interface

The invocation interface table

Information is transmitted between this invocation EXEC and SDF II by means of the invocation interface table, which is an SDF2/DM table with the default name DGIIFTAB. The invocation interface table is a temporary table defined with SDF2/DM table services. It must be open when SDF II is called. Each row in the table identifies one request to SDF II. SDF II can handle multiple rows in one table and will process the requests one after the other, without returning control to the calling program. Only when the end of the SDF2/DM table is reached is control returned to the calling program. Depending on the request, SDF II may add rows to the table.

Two types of errors may occur in the table: row errors and invalid parameters. Rows that cannot be interpreted are ignored. When invalid parameters are passed, the appropriate dialog panel is displayed with the appropriate error message.

In full-screen operation, values that are not found in the DGIIFTAB table are taken from the profile. In batch operation, specify all required parameters.

Figure 38 lists the columns of the invocation interface table and their contents. Unless otherwise indicated, a column applies to all types of requests.

Figure 38 (Part 1 of 4). The columns of the invocation interface table (DGIIFTAB)

Column	Length	Description
IIVROW	1	The type of the row: V To identify a request row. When SDF II reads sequentially through the SDF2/DM table, it performs the requests of only these rows. Rows with request types other than V are ignored. Z To identify a row that has been added by SDF II after successful completion of a request. These rows can be used by the invoker for further processing. They identify the action taken by SDF II. In response to a request, SDF II may add any number of rows. These rows are discussed in more detail on page 64.
IIVREQ	1	The type of request to be processed: A Construct a panel. C Copy an object. D Delete an object. E Edit an object or prototype. F Modify a panel. G Generate an object. M Import an object. O Print the online reference. P Print an object. T Test an object or prototype. X Extract a panel.
IIVNAM	8	The name of the object to be processed.

Figure 38 (Part 2 of 4). The columns of the invocation interface table (DGIIFTAB)

Column	Length	Description
IIVTYP	1	The type of object to be processed: P Panel G Panel group S Partition set A AID table B Control table W Prototype The type of object to be imported: 1 CICS/BMS macros 2 Reserved 3 ISPF panel 4 Reserved 5 Reserved 6 SDF/CICS dump data 7 External source format This column is not used for requests to print the online reference.
IIVLIB	1 or 16	This identifies the library in which the object is stored or is to be stored: • If the library identifier is a single character, SDF II assumes that it is the library identifier assigned in the Specify Libraries dialog. • Otherwise, SDF II assumes that it is a library name of the form: *library***.***sublib* such as PROJECT.GROUP, which must be allocated in the job control statements. In this case, SDF II looks for the 1-digit library identifier defined in the Specify Libraries dialog. For some requests, such as to generate or print an object, the library identifier can be an asterisk (*). This column is not used for requests to edit a prototype, test a prototype, or print the online reference.
IIVONAM	8	This field has a different meaning for each of these requests: **EDIT** A second object name. When you create a new object, this is the name of the skeleton object. If you do not want to use a skeleton object, specify the value X'00'. **COPY** The name of the new object. **MODIFY** The name of the new object. **CONSTRUCT** The name of an SDF2/DM table that contains the panel elements to be used. The default name is DGIU5TAB. See Figure 39 on page 65 for the structure of this SDF2/DM table.

Figure 38 (Part 3 of 4). The columns of the invocation interface table (DGIIFTAB)

Column	Length	Description	
IIVOLIB		This field has a different meaning for each of these requests:	
	1 or 16	**EDIT**	The identifier of the second library: • If the library identifier is a single character, SDF II assumes that it is the library identifier assigned in the Specify Libraries dialog. • Otherwise, SDF II assumes that it is a library name of the form: *library.sublib* such as PROJECT.GROUP, which must be allocated in the job control statements. In this case, SDF II looks for the 1-digit library identifier defined in the Specify Libraries dialog. When you create a new object, this is the library identifier of the skeleton object. If you do not want to use a skeleton object, specify the value `X'00'`.
		COPY	The library identifier of the new object.
		MODIFY	The library identifier of the new object.
	15	**CONSTRUCT**	The 15 characters of the column are made up of: • 3 characters for the depth of the panel in lines. It defaults to the depth defined in the device table. • 3 characters for the width of the panel in characters. It defaults to the width defined in the device table. • 1 character, which specifies: 0 Include the command line in the panel. 1 Exclude the command line from the panel. This applies only to ISPF panels. • 8 characters for the help panel name. This applies only to ISPF, CSP/AD, and VisualGen panels.
IIVDEV	8	This is the device type for these requests:	
		EDIT	To be used when you create a new object without using a default object. It is optional for ISPF.
		CONSTRUCT	For any target system except ISPF, where it is optional.
		EXTRACT	To select the formats to be processed.
		MODIFY	To select the formats to be processed.

Figure 38 (Part 4 of 4). The columns of the invocation interface table (DGIIFTAB)

Column	Length	Description
IIVMOD	1 or 2	This field has a different meaning for each of these requests: **EDIT** The first character holds the option number of the first editor dialog to be invoked. The second character is ignored. **GENERATE** The first character holds the target system for which the object is to be generated: 0 Application prototype 1 CICS/BMS 3 ISPF 5 CSP/AD or VisualGen The second character is ignored. **PRINT** The first character holds the destination of the SDF II output: 1 Standard printer (SYSLST) 4 DBCS printer 5 DBCS printer with field outlining The second character is ignored. **COPY** The type of copy operation: 00 Copy object 10 Copy and replace object 01 Copy object and related objects 11 Copy and replace object and related objects **DELETE** The type of delete operation: 00 Delete object 01 Delete object and related objects
IIVPRO	1	This column controls whether a dialog is to be presented to the user or to be processed immediately: B Do not display any panel of the dialog. G Do not display the panels listed below. P Display these panels with all their values inserted. G and P apply to the following: **EDIT** Identify Object panel or Identify Prototype panel **PRINT** Specify Print Utility Parameters panel **GENERATE** Identify Object for Generation panel and Specify Generation Parameters panel **CONSTRUCT** Specify Panel Construction Utility Parameters panel and Specify Panel Elements panel

Note: To set any of these columns to blank, specify the value `X'00'`. In full-screen operation, you can specify a null value (`''`) for any of the columns IIVNAM, IIVLIB, IIVONAM, IIVOLIB, or IIVDEV. The value from the profile is used instead.

When SDF II has successfully completed a request, it may add one or more rows to the SDF2/DM table, or it may add none. These rows are identified by a row type of Z. They immediately follow the request row, which is identified by a row type of V. The contents of these rows depend on the request, as follows:

Edit Each time you save the object you are editing, a row is added to the SDF2/DM table. This row contains the fields:

IIVNAM The name under which the object was saved
IIVTYP The type of the object
IIVLIB The library in which the object was saved

Copy A row is added that contains the fields:

IIVNAM The name of the target object
IIVTYP The type of the object
IIVLIB The library to which the object was copied

Delete A row is added that contains the fields:

IIVNAM The name of the deleted object
IIVTYP The type of the object
IIVLIB The library from which the object was deleted

Generate For each output file created by the generation utility, a row is added to the SDF2/DM table. This row contains the fields:

IIVNAM The name of the generated object
IIVTYP The type of the object
IIVLIB The library in which the generated object was saved
IIVONAM The member name of the file containing the generated output
IIVOLIB The library containing the generated output

Construct
Each time you save the panel that you are constructing, a row is added to the SDF2/DM table. This row contains the fields:

IIVNAM The name of the constructed object
IIVTYP The type of the object; it is always P (for panel)
IIVLIB The library in which the constructed object was saved

Import Each time an object is saved during import, a row is added to the SDF2/DM table. This row contains the fields:

IIVNAM The name of the imported object
IIVTYP The type of the object
IIVLIB The library in which the imported object was saved

Modify A row is added that contains the fields:

IIVNAM The name of the target object
IIVTYP The type of the object
IIVLIB The library to which the object was copied

No entries are added to the table in response to requests to print the online reference, to print an object, or to test an object or prototype.

Using predefined panel elements

To use predefined panel elements (also called *related fields* in, for instance, the Define Field dialog), you need two user exit routines that supply information about data structures used for your data.

- Retrieve field (DGIUXRET.PROC)

  This user exit routine is used by the panel construction utility or by the Define Fields dialog of the panel editor to retrieve additional information, such as a field's length or occurrence number. See "Retrieve field" on page 67.

- Expand field (DGIUXEXP.PROC)

  This user exit routine is used by the panel construction utility to handle the expand (**ex**) line command and automatic expansion. See "Expand field" on page 68.

Sample routines are available as members DGIUXRET.Z and DGIUXEXP.Z in the SDF II installation library. These routines are intended to be used with the construction utility procedures described in *SDF II General Introduction*. To keep the sample simple, the data to be retrieved is stored in member DGIUX.Z in the SDF II installation library. You would normally retrieve the information from a database.

Modifying the construction utility table

For panel construction, additional information is transmitted to SDF II by means of the SDF2/DM construction utility table, whose default name is DGIU5TAB. You can supply this name, or any name you choose, in the column IIVONAM of the invocation interface table (default name is DGIIFTAB).

The columns of the construction utility table contain basically the same information as the lines of the Specify Panel Elements panel. Column IUV5TYP, however, is not displayed on this panel. It is used only by the invocation interface.

For more information, refer to the SDF II online reference.

Figure 39 lists the columns of the construction utility table and their contents.

Figure 39 (Part 1 of 2). The columns of the construction utility table (DGIU5TAB)

Column	Length	Description
IUV5TYP	1	This is the type of the panel element. The following values are possible: I — This identifies an ISPF procedural section line. Only column IUV5DES is used, which then contains the procedural section line. Table columns of this type are added to the ISPF procedural section of the panel. This is valid for the ISPF target system only. E — Identifies a panel element. All of the following columns are used.
IUV5RFD	≤65	This is the name of the related field, which is passed to the user exit routines (see section "Using predefined panel elements"). You can enter qualified names, such as COLLECT.DATE. Each part of the name can be up to 32 characters long.

Figure 39 (Part 2 of 2). The columns of the construction utility table (DGIU5TAB)

Column	Length	Description
IUV5NAM	31	This is the field name used in SDF II. If left blank, it defaults to the last level of the qualified name in the IUV5RFD column. The default name must be unique within the panel and must conform to the syntax of the programming languages that SDF II supports. Otherwise, SDF II leaves this column blank.
IUV5FFO	8	This is the field format. The following values are possible: EBCDIC The field contains only EBCDIC characters. DBCS The field contains double-byte character set (DBCS) characters. MIXED The field contains EBCDIC and DBCS characters enclosed by SO/SI characters. MSUPRESS The same as MIXED, but SO/SI characters are suppressed (no character position is required for them on printers). DEFMIXED The device default is used. The default value depends on the available markers and on the device type.
IUV5LEN	4	This is the length of the data field. It is optional.
IUV5OCC	3	This is either the occurrence number of a repeat format or the dimension of an array. It is optional.
IUV5VER	1	This is the direction of the array or repeat format. If no occurrence number is specified, it must be blank. The following values are possible: 1 For a vertical array or repeat format. *blank* For a horizontal array or repeat format. This is the default.
IUV5PRO	≤20	This is the prompt text associated with the field.
IUV5DES		This column contains:
	80	For IUV5TYP=I, the contents of an ISPF procedural section line.
	64	Otherwise, the description of the field.

Retrieve field

The retrieve field user exit routine is used to bring into SDF II predefined panel elements from, for example, a database.

Input

The following variable is available in the shared pool:

Figure 40. DGIUXRET input variable

Name	Length	Description
IUVUERFD	256	The name of the related field: • The maximum length of a fully qualified related field name is 256 characters • The name may be qualified • The separator is a period • The maximum length of a name and each qualifier is 32 characters • The name and the qualifiers must not contain blanks

Invocation

The user exit routine DGIUXRET.PROC is called by SDF II.

Output

The user exit routine may set the following variables in the shared pool:

Figure 41. DGIUXRET output variables

Name	Length	Description
IUVUELEN	4	The length of the field (values 0 through 9999)
IUVUEFFO	4	The field format
IUVUEOCC	3	The occurrence number (values 0 through 999)
IUVUEPRO	20	The field prompt
IUVUEDES	64	The field description
IUVUEINI	256	The initial value for the field
IUVUEMSG	8	The identifier of a private message that SDF II displays if an error is detected (RC=16)

Note: See “Modifying the construction utility table” on page 65 for an explanation of the corresponding fields.

Return codes

The return codes from DGIUXRET.PROC are:

0	The related field was successfully retrieved.
16	The related field was not retrieved. The reason is given in the message whose identifier is specified in IUVUEMSG.

SDF II issues message number DGIUM571 for RC=16. To issue a private message instead:

1. Assign the number of the private message to the variable IUVUEMSG in the exit procedure DGIUXRET.PROC.

2. Define the message in SDF2/DM message format. See *SDF II Run-Time Services*.

Expand field

For a field for which the expand (**ex**) line command is entered or for which an automatic expansion is performed, the user exit routine DGIUXEXP is called.

SDF II performs an automatic expansion if you use the invocation interface to invoke the panel construction utility. A panel element is expanded to the next level provided there is one and provided the resulting panel elements are correct. This process is repeated until no further expansion is possible because the lowest level is reached.

Input

The following variable is available in the shared pool:

Figure 42. DGIUXEXP input variable

Name	Length	Description
IUVUERFD	256	The name of the related field: • The maximum length of a fully qualified related field name is 256 characters. • The name may be qualified. • The separator is a period. • The maximum length of a name and each qualifier is 32 characters. • The name and the qualifiers must not contain blanks.

Invocation

The user exit routine DGIUXEXP.PROC is called by SDF II.

Output

The user exit routine creates the SDF2/DM table DGIUXEXP, which may be permanent or temporary. One row is added each time the field is expanded. The columns of DGIUXEXP are:

Figure 43. DGIUXEXP output variables

Name	Length	Description
IUVUNAM	32	The SDF II field name
IUVUELEN	4	The length of the field (values 0 through 9999)
IUVUEFFO	4	The field format
IUVUEOCC	3	The occurrence number (values 0 through 999)
IUVUEPRO	20	The field prompt
IUVUEDES	64	The field description
IUVUEINI	256	The initial value of the field
IUVUEMSG	8	The identifier of a private message that SDF II displays if an error is detected (RC=8 or RC=16)

Note: See “Modifying the construction utility table” on page 65 for an explanation of the corresponding fields.

Return codes

The return codes from DGIUXEXP are:

0	The related field was successfully expanded.
8	Expansion is not possible because the last level of the field was reached.
16	The last level was not reached, and the related field was not expanded. The reason is given in the message whose identifier is specified in IUVUEMSG.

SDF II issues message number DGIUM571 for RC=16 and message number DGIUM572 for RC=8. To issue private messages instead:

1. Assign the number of the respective private message (RC=8 or RC=16) to the variable IUVUEMSG in the DGIUXEXP.PROC.
2. Define the messages in SDF2/DM message format. See *SDF II Run-Time Services*.

Using generated output

This section describes the types of generated output and how to further process them. The generated output depends on the target system that you have specified for the generation on the Identify Object for Generation panel.

Prototype The generated output is an SDF2/DM panel with additional information for application prototyping. Use it in the SDF II prototype facility to simulate an application. Refer to *SDF II Run-Time Services*.

Refer to *SDF II General Introduction* for what you have to do before you can use the SDF II prototype facility.

CICS/BMS Depending on what you specified on the Specify Generation Parameters panel, the generated output consists of either or both:

- CICS/BMS macros, created from SDF II panels, panel groups, and partition sets. Process them in the same way that you would process macros that were not created by SDF II but with, for example, a standard editor.
- Application data structures. Include them in your source program.

ISPF Depending on what you specified on the Specify Generation Parameters panel, the generated output consists of either or both:

- An ISPF panel. Put this panel directly into the panel library of your application program.
- Application data structures. Include them in your source program.

  Besides the data structure for your application program, data structures are generated for use within ISPF VDEFINE and VDELETE service calls. You must code the ISPF service calls yourself. Skeletons are provided as examples of these calls.

  The generated data structures represent the parameters to the VDEFINE/VDELETE service calls for a panel. It is assumed that all variables of a panel are defined by a single service call with the LIST option.

CSP/AD The generated output is CSP/AD external source format.

Use the import utility of CSP/AD to import the external source format into CSP/AD.

VisualGen The generated output is CSP/AD external source format.

Use the import utility of VisualGen to import the external source format into VisualGen.

Writing a generation user exit routine

When you generate an object, you can specify the name of a user exit routine to be invoked after successful generation.

For CICS/BMS or ISPF, you can use two different user exit routines: one for application data structure generation, the other for CICS/BMS macro generation or ISPF panel generation. For CICS/BMS panels, the user exit routine for CICS/BMS macro generation is invoked after generation for each format instance.

For CSP/AD, you have only one user exit routine.

All user exit routines are REXX EXECs. They can access a set of variables stored in the shared pool. They can be used to further process the generated output. For example, they could submit a job to assemble the generated CICS/BMS macros.

The following variables are available to the user exit routine:

Figure 44 (Part 1 of 2). Generation user exit routine

Column	Length	Description
ICVUENAM	8	The name of the object that has been generated.
ICVUETYP	1	The type of object that has been generated. The following values are possible: G Panel group P Panel S Partition set Y Prototype
ICVUELIB	1	The identifier of the library in which the generated object is stored.
ICVUETSY	1	The target system for which generation is done. The following values are possible: 0 Generation for application prototype 1 Generation for CICS/BMS 2 Reserved 3 Generation for ISPF 4 Reserved 5 Generation for CSP/AD or VisualGen
ICVUEGTY	1	The type of generation. The following values are possible: D A data structure has been generated. C CICS/BMS macros, ISPF panels, or a CSP/AD external source format have been generated.

Figure 44 (Part 2 of 2). Generation user exit routine

Column	Length	Description
ICVUELAN	1	The programming language of the generated data structure. The contents of this variable are undefined when the value of ICVUEGTY is C, or when generating for a target system other than CICS/BMS or ISPF. The following values are possible: A Assembler C COBOL P PL/I R RPG II (this is supported only for CICS/BMS) X C language
ICVUEOLN	16	The name of the VSE sublibrary.
ICVUEOMN	8	The name of the member containing the generated output.
ICVUEOVT	8	The type of the member containing the generated output.

SDF II provides three sample user exit routines in its installation library:

DGICXBRW.PROC	Displays the generated output in a scrollable list on an SDF II panel.
DGICXBUD.PROC	Displays the generated output by means of the DITTO/ESA utility.
DGICXASM.PROC	Assembles a CICS/BMS panel object.

Getting the bill of materials table

SDF II creates an SDF2/DM table (DGICVBOM), which contains all objects that were referenced during generation.

The following variables are available:

Figure 45. Bill of material table

Column	Length	Description
IIVBOMN	≤8	The name of the referenced object.
IIVBOMT	6	The type of the referenced object. The following values are possible: DGIPNL For a panel DGIGRP For a panel group DGIPST For a partition set DGITBL For an AID table DGIOCT For a control table
IIVBOML	1	The identifier of the library in which the object is stored.

Specifying a user exit for the data extraction and modification utilities

The sample DGIUX60C.Z in the installation library shows how to specify a user exit for the data extraction and modification utilities.

Variables the user exit routines can access

Figure 46 lists the variables that are set in the shared pool before the user exit routine is called.

Figure 46 (Part 1 of 2). General output variables

Name	Length	Description
The following shared pool variables are set before processing of an object starts:		
IUV62FUN	1	Function: **1** Extract data. **2** Copy object and modify contents.
IUV62NAM	8	The name of the object.
IUV62LIB	1	The library ID from where the object has been read.
IUV62TYP	1	The type of the object: **P** Panel
IUV62NNA	8	The name of the new object.
IUV62NLI	1	The library ID where the new object will be stored.
The following variable is set before each user exit routine call:		
IUV62REQ	3	The type of the item to process: PPB Beginning of panel processing PPE End of panel processing PFB Beginning of format processing PFE End of format processing PEV Variable field PEC Constant field PEB Background PEP Pull-down choice
The following variables are set before the user exit routine is called with IUV62REQ='PFB':		
IPVFODVT	8	Device type of the format instance.
IPVFOFON	8	Format name (applicable only for MFS panels).
IPVFOPNB	*num*	Number of physical pages (applicable only for MFS panels, it is 1 for all other target systems).
IPVFOPNN	*num*	Index number of physical page (1 ... IPVFOPNB) (applicable only for MFS panels, it is 1 for all other target systems).
IPVFODEP	*num*	Format depth. It is 0 if the device name (IPVFODVT) identifies a stream device.
IPVFOWID	*num*	Format width.
The following variables are set when the user exit routine is called with IUV62REQ='PE*x*':		
IPVEONAM	32	The name of the field. It is blank for background items.
IPVEOLIN	*num*	Vertical position of the item. The value of *num* is 0 if the device name (IPVFODVT) identifies a stream device.
IPVEOCOL	*num*	Horizontal position of the item.
IPVEOINC	8	If the item is part of an include panel this field contains the name of the include panel. **Note:** No changes can be performed within the include panel. The include panel has to be processed as separate object.

Figure 46 (Part 2 of 2). General output variables

Name	Length	Description
IPVEOLIX	8	If the item is part of a repeat format this variable contains the index. Otherwise it is 0.
IPVEOARX	8	If the field is part of an array this variable contains the index. Otherwise it is 0.
IPVEOFFO	1	Field format of the item: E EBCDIC M Mixed D DBCS (possible only for fields) S Mixed suppress (MFS panels only) Background can only be EBCDIC or mixed.
IPVEOLEN	*num*	The length of the item: • For fields this is the length of the field • For background this is the length from the first position (IPVEOCOL) up to the next field or to the end of the line, whichever is shorter
IPVEOTXT	any	The text associated with the item. It contains: • For variable fields, the initial value, or the literal in case of MFS panels • For constant fields and for background, the associated text The following rule applies to the text: • For background this is the text from the first position (IPVEOCOL) up to the next field or to the end of the line, whichever is shorter

Additional variables the user exit routine for the copy utility can access

Figure 47 lists variables that can be modified by the user exit routine within the shared pool.

Figure 47. Input variables

Name	Length	Description
IPVENFFO	1	Field format of the item. It can be set for variable and constant fields and for the background when the device the instance is defined for supports it. E EBCDIC M Mixed D DBCS (possible only for fields) S Mixed suppress (MFS panels only) (Background can be only EBCDIC or mixed.)
IPVENTXT	any	The text associated with the item. It contains: • For variable fields, the initial value text • For constant fields and for background, the text

Chapter 6. Importing objects into SDF II

This chapter describes how to use the import utility to import objects from various sources into SDF II libraries.

Note: The IBM licensed programs from which object definitions can be imported into SDF II are:

- SDF/CICS OS/VS Release 5.0 (5740-XYF)
- SDF/CICS VSE Release 5 (5746-XXT)
- SDF/CICS CMS Release 1.0 (5664-178)

The following types of object definitions can be imported into an SDF II-supported library:

- Maps, map sets, and partition sets defined with CICS/BMS macros.
- Panels defined in SDF2/DM panel syntax.
- Maps and map groups defined in external source format and exported from CSP/AD or VisualGen (see "Importing external source format" on page 81).
- Maps and map groups defined in extended external source format (see "Extended external source format" on page 81).
- Maps, map sets, and partition sets defined and dumped with SDF/CICS.

Select the import utility, which is option **9.2** on the Select an SDF II Function panel or option **2** on the Select a Utility panel. The Specify Import Utility Parameters panel is then displayed. Specify, on this panel, the member containing the objects to be imported and the library in which the imported objects are to be stored.

Importing objects from CICS/BMS

Use the import utility to create SDF II panel, panel group, and partition set objects from CICS/BMS macro statements. The import utility imports one file of map set macros or partition set macros. The utility assumes that the input consists of correct macros.

The following checks are made:

- The first macro statement is checked to make sure that it is either a DFHMSD macro for map sets or a DFHPSD macro for partition set definitions.
- Macro parameters are checked syntactically and, if any required parameters are missing, a message is issued.

Comments and remarks on the macro statements are ignored, unless they are data structure name definition records. These records, which can be added as comment lines to the CICS/BMS macros to be imported, are used by the import utility to build data structure names. For the format of the data structure records see "Importing data structure names" on page 75.

If an object with the same name already exists, a new instance is created if the current device is different from the devices for which the existing object is defined. An SDF II object is created only if the macro statements are correct.

Panel and panel group import

SDF II panels and panel groups are created from CICS/BMS DFHMSD, DFHMDI, and DFHMDF macro statements.

The input is a file of CICS/BMS macro statements. These statements define a map set that starts with a DFHMSD statement. The DFHMSD statement contains one or more maps that start with a DFHMDI statement. The last statement must be DFHMSD TYPE=FINAL.

One panel group is created for each map set; one panel is created for each map.

Partition set import

An SDF II partition set is created from CICS/BMS DFHPSD and DFHPDI macro statements.

For any missing optional parameters, the SDF II partition set editor defaults are used.

Checks are made that:

- The viewports are correctly positioned in the usable area, and that the size of the viewports is correct.
- Partitions do not overlap.
- For the IBM 8775 display terminal, the partition buffer, depth buffer, and scroll storage limits are correct.

Importing data structure names

Use the import utility to create data structure names from records that you add as *DGI comment lines to CICS/BMS macros to be imported.

During import, SDF II uses the information in these records to assign data structure names to imported fields. The imported data structure names will later be used for the generation of the panel's data structures.

You can define data structure names for fields, arrays, CICS/BMS groups, and subfields for CICS/BMS. For CICS/BMS maps and map groups, it is also possible to create arrays and repeat formats using data structure name records.

These types of records are necessary to create a data structure name definition:

*DGI*DSECT	DSECT record.
DGI	One or more element records.
*DGI*REP*	Repeat element record; optional. It is used to create a repeat format in the imported SDF II panel.
*DGI*END	DSECT end record.

The structure of one data structure name definition is as follows:

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
*DGI*DSECT mmmmmmmm                                                     00000010
*DGI*lv length occ dsect_name                                           00000020
*DGI*...                                                                000000..
*DGI*REP*occ row col dep wid dsect-name                                 000000..
*DGI*...                                                                000000..
*DGI*END                                                                00000090
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
```

The element records that define one data structure must not be interrupted by any lines, except for comment lines or blank lines, which are ignored by SDF II.

Figure 48 shows the structure of the DSECT record:

Figure 48. DSECT record

Columns	Description
1–10	*DGI*DSECT
11	blank
12–19	Map name
20–72	blank
73–80	Sequence number

Figure 49 shows the structure of the **element record**.

Figure 49. Element record

Columns	Description
1–5	*DGI*
6–7	A 2-digit level number. Possible values are: 01 Reserved 02 BMS field, BMS group, BMS array 03 BMS subfield in a group BMS field, BMS group, BMS array within a repeat format 04 BMS subfield of a group within a repeat format
8	blank
9–14	Length of field: 1–256. It must be blank for BMS groups.
15	blank
16–18	Occurrence number: 1–999 or blank
19	blank
20–71	Name of the data structure element or blank. (SDF II currently supports names up to 31 characters long.)
72	blank
73–80	Sequence number

Figure 50 shows the structure of the **repeat element record**.

Figure 50. Repeat element record

Columns	Description
1–9	*DGI*REP*
10–12	Occurrence number of repeat format (1–999)
13	blank
14–16	Row number in which the repeat format is to start (1–999)
17	blank
18–20	Column number in which the repeat format is to start (1–999)
21	blank
22–24	Depth of the repeat format (1–999)
25	blank
26–28	Width of the repeat format (1–999)
29	blank
30–71	Name of the repeat format or blank. (SDF II currently supports names up to 31 characters long.)
72	blank
73–80	Sequence number

Rules for CICS/BMS

When you import from CICS/BMS, observe the following:

- The fields of one or more maps can be named. One data structure name definition is required per map.
- Each definition must be within the same file as the CICS/BMS macros defining the map.
- A data structure name definition for a map can be specified at any point in the file if a map name is specified in the begin record.
- If no map name is specified in the begin record, the begin record will be associated with the map defined by the previous DFHMDI macro. Such a definition cannot be placed before the first DFHMDI macro.
- One element record must be specified for each variable field, CICS/BMS group, or subfield.

  The sequence of element records must be the same as the sequence of the corresponding DFHMDF macros of the variable fields, groups, or subfields.

  The element record for a CICS/BMS group must be specified immediately before the element record that describes the first subfield of this group (a CICS/BMS group has no associated DFHMDF macro).
- The level must be:

  02 BMS field, BMS group, BMS array
  03 BMS subfield in a group
  BMS field, BMS group, BMS array within a repeat format
  04 BMS subfield of a group within a repeat format

- The length:
  - Must not be specified for CICS/BMS groups.
  - Is optional for fields and subfields. If specified, it must match the value specified in the DFHMDF macro.
- The occurrence number for arrays:
  - Must be specified and must match the value *n* in OCCURS=*n* if specified in the DFHMDF macro.
  - Can be specified to combine fields or BMS groups into an array. The fields that will be combined into an array:
    - Must be in sequence, without any DFHMDF macros, not belonging to the array, in between
    - Must have the same subfield structure, that is, if the first field is a BMS group, each field to be added to the array must be a BMS group with the same subfield structure as well
    - Must have the same length and no occurrence number
    - Must be identically defined in justification, case, input/output pictures, and decimal positions.
  - Can be specified to combine subfields of a group into an array. The subfields that will be combined into an array:
    - Must all be part of the same BMS group
    - Must be in sequence, without any intervening DFHMDF macros that do not belong to the array
    - Must have the same length
- To create a repeat format:
  - Add a repeat element record before the first record that is to be part of the repeat format. The repeat element record must specify the position (row, column), size (depth, width), and repetition factor (occurrence number) of the repeat format. The repeat format must not wrap out of the panel format.
  - All fields that are part of the repeat format have a level that is one higher than for the same construct outside the repeat format:

Construct	Level within	Level outside
Fields, BMS groups, arrays	3	2
Subfields	4	3

  - The structure of the repeat format must be identical in the base part and each repeated part, that is:
    - The number of fields must be the same
    - The sequence of fields must be the same
    - If a field is a BMS group in one part, it must be a group with identical subfield structure in all other parts as well
    - If a repeat format part contains an array, the same array must exist in all other parts as well
  - Repeated fields must have the same:
    - Size
    - Occurrence number
    - Field format (EBCDIC or mixed)

    - Application attributes (justification, case)
    - Input/output pictures, decimal positions
  - A repeat format cannot contain another repeat format.
  - A field that is part of the repeat format must be fully contained in a repeat format part. This includes also the attribute byte in front of the field.
  - Constant field text may vary in the repeated parts.

Importing objects from ISPF

Use the import utility to create an SDF II panel from an ISPF panel. The ISPF panel must be syntactically correct. If the ISPF panel has not been used previously in an ISPF application, it should have been checked in PDF test mode.

The import utility builds a Marks table that contains at least the definition of the background attributes, a constant field mark, and a variable field mark.

The import utility handles the information in the various sections as follows:

- Information from the NUMBER parameter of the)CCSID section will be stored in the ISPF characteristics.
- If parameters are specified in the)PANEL section, the information they contain will be stored in the ISPF characteristics.
- If an)ATTR section is defined, attribute characters are translated into marks, if possible; otherwise, they are translated into attribute descriptors.

  CUA panel element attribute characters of the type action bar or reference phrase are always translated into marks.
- The information from the)ABC,)ABCINIT, and)ABCPROC sections is stored with the corresponding fields.

  All .ZVARS statements are removed from the)ABCINIT sections; the names contained therein become pull-down field names.
- The)BODY section is translated into the format part of the SDF II panel. ISPF INPUT and OUTPUT fields are translated into variable fields. The length of a variable field is determined by the position of the attribute following the field.

  CUA panel element fields are translated as follows:

  - Fields of the types CEF, LEF, NEF, EE, VOI, LI, and LID are translated into variable fields.
  - Fields of the types AB, ABSL, CH, CT, ET, PT SAC, SI, SUC, WASL, WT, and RP are translated into constant fields.
  - Fields of the types DT, FP, NT, and PIN are translated either into constant fields or into background text.

  Text fields are translated into either constant fields or background text. The background text is used when the field has normal intensity and no other attribute has been defined.

  The dynamic and graphic areas are unchanged.
- Any)MODEL section is translated into a repeat format with an occurrence number of 1 or *.

- For the)AREA section, a separate format is created, which will show only the contents of the scrollable area: it will not show elements of the main format. In the main format, only the window in which the scrollable area is to be displayed will be shown by using the appropriate area mark.
- If an)INIT section is defined, SDF II processes the statements it contains, as follows:
  - Any .ZVARS statement is removed from the)INIT section; the names contained therein become field names.
  - Any .CURSOR and .CSRPOS statements are translated into the cursor attribute.
  - Any .HELP statement is kept as the help panel name in the general panel characteristics.
  - An initialization statement (as in the assignment of a literal to a field variable that has no value) is removed from the)INIT section. The literal is then translated into an initial field value. For example, the code:

    ```
    IF(&FIELD = ' ')
       &FIELD = 'value'
    ```

    is removed from the)INIT section. *value* then becomes the initial value of &FIELD.

  Any other statements of the)INIT section are added to the list of)INIT section statements.
- All statements of the)REINIT section remain unchanged and are added to the list of)REINIT section statements.
- All statements of the)PROC section remain unchanged and are added to the list of)PROC section statements.
- The information from the)HELP section is stored with the corresponding field.
- All lines of the)END section remain unchanged and are added to the list of)END section lines.

Restrictions

- In the)INIT section, the)REINIT section, and the)PROC section of an ISPF panel, more than one statement can appear on each line.

  Any of the following statements of the)INIT section must be the only statement on a line if it is to be correctly imported:

  .ZVARS
  .CURSOR
  .CSRPOS
  .HELP
  An initialization statement.
- & variables in initial values are not resolved.
- Fields of the type **Input** or **Output** or equivalent CUA-type fields in a scrollable area must not:
  - Start in column 1 or end in the last column
  - Wrap around lines

Importing external source format

Use the import utility to create panels and panel groups from map and map group structures originally created in CSP/AD external source format or in extended external source format. (For information about extended external source format, see “Extended external source format.”)

The import utility uses the external source format as input and creates SDF II objects.

To create an external source format, use the CSP/AD export utility, VisualGen export utility, or other tools that support this format.

For information about standard external source format refer to *CSP/AD External Source Format Reference*.

Extended external source format

SDF II uses extended external source format tags and their attributes to represent panel properties that are supported in CICS/BMS and IMS/MFS. Extended external source format tags are not supported in either CSP/AD or VisualGen.

For each standard map or map group structure tag, there is a corresponding extended tag. Within an extended tag, all standard and extended attributes are allowed. Within standard tags, no extended attributes are allowed. Within one map or map group structure, standard tags and extended tags cannot be mixed.

If the map or map group structure contains only standard tags, a panel or panel group is created for target system CSP/AD or VisualGen.

If the map or map group structure contains extended tags, a panel or panel group is created for target system ALL.

The following list shows the extended tags and corresponding standard external source format tags:

Extended Tag	Standard Tag
DGIAREA	AREA
DGICATTR	CATTR
DGICFLD, EDGICFLD	CFIELD, ECFIELD
DGIPRSNT	PRESENT
DGIMAP, EDGIMAP	MAP, EMAP
DGIMAPED	MAPEDITS
DGIMAPG, EDGIMAPG	MAPG, EMAPG
DGIMSGS	MESSAGES
DGIVFLD, EDGIVFLD	VFIELD, EVFIELD
DGIVATTR	VATTR
DGIAID, EDGIAID	
DGIPFK	
DGIMFLD, EDGIMFLD	

The extended tags DGIAREA, DGIPRSNT, DGIMAPED, DGIMAPG, and DGIMSGS have the same attributes as the corresponding standard external source format tags.

The extended tags DGICATTR, DGICFLD, DGIMAP, DGIVATTR, and DGIVFLD have attributes in addition to those of the corresponding standard external source format tags. These tags are listed here:

DGICATTR and **DGIVATTR**

The following attributes are supported for target system IMS/MFS only:

SLDI = *nn*

The line density in lines per inch. The minimum is 1; the maximum is 72.

SLDP = *nn*

The line density in points per inch. The minimum is 1; the maximum is 72.

DGICFLD

The following attribute is supported for target system IMS/MFS only:

ATTR = y | n

y Specifies that the position in front of this field is reserved for attributes. The external source format field position is the position of the attribute.

n Specifies that the position in front of this field is not reserved for attributes. The field position is the position that would have been used by an attribute. Specification of COLUMN=0 leads in this case to a field with the first field position in column 1. Do not specify **n** unless it is supported by the device.

The default depends on the device type. It is **n** if the device does not require a separate byte position for the attribute. Otherwise, the default is **y**.

DGIMAP

The following attributes are supported for target systems CICS/BMS and IMS/MFS:

ADJUNCT = y | n

Specifies the types of adjuncts to be generated:

y Specifies that adjuncts for dynamic modification of all attributes are to be generated in the data structure for each field in the map.

n Specifies that only those adjuncts required by the target system are to be generated. The default is **n**.

CURSOR = *line column*

Specifies the position where the cursor is to be placed initially.

DEVICES

Within a DGIMAP tag, in addition to the valid CSP/AD or VisualGen devices, the following device names are supported:

- 5550
- All 3270-like devices supported by CICS/BMS and IMS/MFS
- 3270P, 3270P1, 3270P2
- SCS1, SCS2

TITLE = '*text*'

A panel description of up to 32 characters within quotes. Quotes of the same kind as the delimiting quote that are included in the text must be doubled; each pair of these doubled quotes counts as one character.

The following attribute is supported for target system CICS/BMS only:

FRSET = y | n
Specifies that the modified data tags of all fields currently in the 3270 buffer are to be reset to a not-modified condition before any map data is written to the buffer:

y Specifies that the modified data tags are to be reset.

n Specifies that the modified data tags are not to be reset. The default is **n**.

The following attributes are supported for target system IMS/MFS only:

EJECT = *bgnpp*|*endpp*|*bgnmsg*|*endmsg*
Printer form feed value.

FORMS = '*literal***'**
A forms literal of up to 16 characters within quotes.

SCA = y |n
Specifies whether the map is to contain a system control area:

y Specifies that the map contains a system control area.
n Specifies that there is no system control area. The default is **n**.

SLDI = *nn*
The line density in lines per inch, from 1 to 72.

SLDP = *nn*
The line density in points per inch, from 1 to 72.

TCLIT = '*literal***'**
Specifies that the map contains a transaction code. The last character of the literal string should be a blank. If it is not, a blank is added. The maximum length is 256.

DGIVFLD
The following attributes are supported for target system IMS/MFS only:

ATTR = y | n
Specifies whether a position is to be reserved for field attributes:

y Specifies that the position in front of this field is reserved for attributes. The external source format field position is the position of the attribute.

n Specifies that the position in front of this field is not reserved for attributes. The field position is the position that would have been used by an attribute. Specification of COLUMN=0 leads in this case to a field with the first field position in column 1. Do not specify **n** unless it is supported by the device.

The default depends on the device type. It is **n** if the device does not require a separate byte position for the attribute. Otherwise, the default is **y**.

SYSLIT = *ltseq*|*ltname*|*time*|*date1*|*date2*|*date3*|*date4*|*lpageno*|*ltmsg*
Identifies the field as a system literal.

SYSMSG = y | n
Specifies whether the field is a system message field:

y Specifies that the field is a system message field.
n Specifies that the field is not a system message field. The default is **n**.

The extended tags DGIAID, DGIPFK, and DGIMFLD have no corresponding standard external source format tags. They are used to represent the information that is defined by the PFK parameter of the DEV utility control statement in IMS/MFS. In SDF II, this information is kept in an IMS/MFS AID (attention identifier) table.

DGIAID
This tag defines an AID table.

AIDNAME = *table_name*
Identifies the AID table.

DATE = *date*
Specifies the date when the AID table was last modified.

TIME = *time*
Specifies the time when the AID table was last modified.

The format rules for the attributes are the same as for their equivalents in the MAP or MAPG tags.

One DGIAID plus one or more DGIPFK form one AID table definition. AID tables can be imported stand-alone or can be included in the map structure. If an AID table is imported within the map structure, the tags for the AID table must be before the associated tags for the map.

DGIPFK
This tag defines one PF key in an AID table.

KEY = *number*
Identifies the number of the PF key to be defined. It must be a number in the range 1–24.

FUNCTION = *function*
Specifies the control function to be assigned to the PF key. These keywords can be specified:

NEXTPP	Page advance
NEXTMSG	Message advance
NEXTMSGP	Message advance protect
NEXTLP	Next logical page
ENDMPPI	End multiple page input

LITLEN = *length*
Specifies the length of the literal. It must be a number in the range 1–256. If DBCS=Y is specified, it must be an even number.

DBCS = y | n
Specifies whether the literal must be DBCS:

y Specifies that the literal must be DBCS.
n Specifies that the literal may be EBCDIC or mixed DBCS.

literal
Is the literal associated with the PF key.

- For DBCS=n, it may be EBCDIC or mixed DBCS.
- For DBCS=y, it must be DBCS. It must start with SO and end with SI without any other SO or SI between. The length of the literal is counted without the SO/SI symbols, which will be removed by SDF II.

- If the value specified for LITLEN is greater than the length of the literal, the literal will be padded with blanks up to the specified length.
- The maximum length of the literal is 256 characters.

DGIMFLD

This tag defines a PFK receiver field and relates it to an AID table. The field name corresponds to the name of the input field defined by the PFK parameter of the DEV utility control statement in IMS/MFS.

NAME = *field_name*

Identifies the name of the field. For the name, the same rules apply as for the NAME attribute of the VFIELD tag.

BYTES = *field_length*

Specifies the field length in bytes. For PFK receiver fields, it must be a number in the range 1–256.

AIDNAME = *aid_table_name*

Relates the field to an AID table from which the program function key literal or control function data is taken. The field will be added as PFK receiver field to the SDF II input data structure only.

Only one PFK receiver field can be defined in a map.

Restrictions

Within one map or map group structure, standard and extended tags must not be mixed.

In CSP/AD and VisualGen, the import utility does not accept extended external source format input.

If more than one device name is specified in the DEVICES attribute, import searches the device table for the smallest device list that contains these devices as a subset. If no device list is found, import for the object stops.

When a panel is created in a library, a panel group is created in the same library if:

- A panel group with the name specified for the GRPNAME attribute of the map structure does not already exist in the library.
- The name specified for the GRPNAME attribute is not the default name (DGIGRP).

If a panel group with the name specified for GRPNAME exists in the same library as the panel, the panel name will be added to the panel list regardless of the target system of the panel group.

If a panel group is renamed, all subsequently created panels for this group are added to the renamed group.

Migrating objects from SDF/CICS

To migrate data from SDF/CICS to SDF II:

1. Define a migration sublibrary.
2. Unload the data from SDF/CICS.
3. Copy the data to be migrated to the migration sublibrary.
4. Import the data to SDF II.

Unloading data from SDF/CICS

You can unload data from SDF/CICS by using either tape or POWER PUNCH DISP=I.

Unloading data from SDF/CICS using tape

Figure 51 shows an example job for unloading data from SDF/CICS to a migration sublibrary. The job dumps data from SDF/CICS to a tape, and copies it into the target sublibrary, using DITTO/ESA function TL Tape to Library. Modify the job to meet the requirements of your installation. If you use different library and member names, modify the jobs illustrated in Figure 51 and in Figure 53 on page 88.

```
* $$ JOB JNM=DGIUJMGT
// JOB DGIUJMGT        migrate SDF/CICS objects
// ON $RC>0 GOTO $EOJ
// SETPARM TAP='480'
// OPTION NOLOG
/. C ------------------------------------------------------------------
/. C     SCREEN  DEFINITION  FACILITY II  -  RELEASE 6 LEVEL 0
/. C          LICENSED MATERIAL - PROGRAM  PROPERTY OF IBM
/. C 5746-XXT (C) Copyright IBM Corp. 1979, 1997.  ALL RIGHTS RESERVED.
/. C Migrate from SDF/CICS to SDF-II for VSE using a tape
/. C This sample assumes all necessary DLBL/EXTENT statements are in
/. C standard labels and the SDF/CICS installation library is in the
/. C permanent search chain.
/. C ------------------------------------------------------------------
/. C Part 1 - unload part(s) to tape using SDF/CICS utility
/. C ------------------------------------------------------------------
// MTC REW,&TAP.
// ASSGN SYS024,&TAP.
// TLBL DVDDUMP,'SDFCICS.DUMP'
// EXEC DVD6BTCH,SIZE=AUTO
*PARM U=user/pwd
UNLOAD OBJECT=*.*,FUNCTION=DUMP
/*
// MTC REW,&TAP.
/. C ------------------------------------------------------------------
/. C Part 2 - copy unloaded object(s) into a migration library
/. C          using DITTO/ESA
/. C          modify the $$DITTO statements to meet the requirements
/. C          of your installation
/. C ------------------------------------------------------------------
// UPSI 1
// EXEC DITTO
$$DITTO TL INPUT=SYS024,FILEIN=DVDDUMP,
$$DITTO   LIBOUT=SDFOBJ.MIGRATE,MEMBEROUT=MIGRATE.DATA
$$DITTO EOJ
/*
/&
* $$ EOJ
```

Figure 51. DITTO/ESA job to unload SDF/CICS dump data

After you have run this job, the exported data is in the member MIGRATE.DATA in sublibrary SDFOBJ.MIGRATE. Figure 53 on page 88 shows an example job for importing the objects.

Unloading data from SDF/CICS using PUNCH DISP=I

Figure 52 shows an example job for unloading the data from SDF/CICS to a migration library.

The job dumps data from SDF/CICS, using DISP=I, and copies it to a sublibrary member, using the catalog command of VSE/Librarian. Modify the job to meet the requirements of your installation. If you use different library and member names, modify the jobs illustrated in Figure 52 and in Figure 53 on page 88.

```
* $$ JOB JNM=DGIUJMGP
* $$ PUN DISP=I
// JOB DGIUJMGP migrate via VSE/POWER queues
// OPTION NOLIST
/. C ------------------------------------------------------------------
/. C     SCREEN  DEFINITION  FACILITY II  -  RELEASE 6 LEVEL 0
/. C          LICENSED MATERIAL - PROGRAM  PROPERTY OF IBM
/. C 5746-XXT (C) Copyright IBM Corp. 1979, 1997.  ALL RIGHTS RESERVED.
/. C Migrate from SDF/CICS to SDF II for VSE using power data queues
/. C with disp=i.
/. C This job punches a librarian catalog job including the object data
/. C from SDF/CICS unload function.
/. C ------------------------------------------------------------------
/. C Part 1 - punch jcl statements
/. C ------------------------------------------------------------------
// EXEC ASSEMBLY
 PUNCH '// EXEC LIBR'
 PUNCH ' AC S=SDFOBJ.MIGRATE'      <== the sublibrary for migration
 PUNCH ' CAT MIGRATE.DATA R=Y'     <== the member to migrate later
 END
/*
/. C ------------------------------------------------------------------
/. C Part 2 - unload part(s) using SDF/CICS utility
/. C ------------------------------------------------------------------
// ASSGN SYS024,SYSPCH
// EXEC DVD6BTCH,SIZE=AUTO
*PARM U=user/pwd
UNLOAD OBJECT=*.*,FUNCTION=DUMP
/*
/. C ------------------------------------------------------------------
/. C Part 3 - punch the rest of the migration
/. C          modify IIVNAM specification below to match the
/. C          library definitions in the above catalog statement
/. C ------------------------------------------------------------------
// EXEC ASSEMBLY
 PUNCH '/+'
 PUNCH '/*'
 END
/*
/&
* $$ EOJ
```

Figure 52. Job to unload SDF/CICS dump data, using PUNCH DISP=I

After you have run this job, the exported data is in the member MIGRATE.DATA in sublibrary SDFOBJ.MIGRATE. Figure 53 on page 88 shows an example job for importing the objects.

Importing the data from the migration sublibrary into SDF II

Figure 53 shows an example job for migrating the library member, using SDF II batch migration. Modify the job to meet the requirements of your installation. If you use different library and member names, modify the jobs illustrated in Figure 53 and in either Figure 51 on page 86 or Figure 52 on page 87.

```
* $$ JOB JNM=DGIUJMGM
// JOB DGIUJMGM
/. C -----------------------------------------------------------------
/. C     SCREEN  DEFINITION  FACILITY II  -  RELEASE 6 LEVEL 0
/. C          LICENSED MATERIAL - PROGRAM  PROPERTY OF IBM
/. C 5746-XXT (C) Copyright IBM Corp. 1979, 1997.  ALL RIGHTS RESERVED.
/. C Part 3 - migrate the library member using SDF II batch migration
/. C          modify ILV10L01 and IIVNAM to match the specifications
/. C          given in the $$DITTO statements above.
/. C -----------------------------------------------------------------
// EXEC PROC=DGIPROC
ILV10L01 = 'SDFOBJ.USER'           /* library 1                     */
IIVREQ='M'                         /* request=import                */
IIVTYP='6'                         /* type=SDF/CICS object          */
IUV20DVL='SDFOBJ.MIGRATE'          /* L.S of input member           */
IUV20DVM='MIGRATE.DATA'            /* M.T of input member           */
IIVLIB='1'                         /* to library 1 specified above  */
/*
/&
* $$ EOJ
```

Figure 53. Job to migrate SDF/CICS dump data to SDF II

For a description of the parameters, see “Example 5: Migrate SDF/CICS dump data to SDF II” on page 47.

Use of the SDF II device table

For SDF/CICS, each device type has an associated SDF/CICS code. When you migrate objects from SDF/CICS, the device table is searched for an SDF II device with a matching SDF/CICS code.

The first device that has a matching code in the SDF II device table is the device type of the created SDF II panel or panel group.

If an SDF/CICS map or map set contains a specification that has more than one device type, the device list in the SDF II device table must contain all the device types of that specification.

SDF II scans the device table for each device type of the specification. It then scans for a device list that contains only these device types. The first device list fulfilling the condition is the device type of the created SDF II panel or panel group.

When you migrate an SDF/CICS partition set, an SDF II device with the same name as the partition set device name must exist. This will be the device type of the created SDF II partition set.

To update the device table, use the device table editor, which you can access in the SDF II customization dialog.

Restrictions

If a format contains array elements in a random sequence (that is, if they are ordered neither horizontally nor vertically), the import utility reorders the array elements into a horizontally ordered array and displays a message. Arrays of structures are migrated as repeat formats, all of whose fields must be completely contained within the repeat format. Arrays of structures that cannot be mapped to a repeat format are ignored.

When SDF II migrates SDF/CICS maps that have the same name but belong to different map sets, it creates duplicate names. These duplicate names are processed as defined in the **Option** field of the Specify Import Utility Parameters panel.

The original names are retained in the description and as the generation names of any migrated object.

If a partition set is defined with more than one suffix for the same partition set device, only the first specification is migrated to SDF II.

For fields of length zero in SDF/CICS, only outlining attributes are migrated to SDF II.

Appendix A. The device table

This appendix lists devices supported by SDF II. The column headings in the table are abbreviated. The following list expands them:

Column heading	**Meaning**
NAM	Device name
DVL	Indication for device list (y=DL, n or blank=no DL)
SDF	SDF/CICS device code
BMS	CICS/BMS device suffix
GDD	GDDM–IMD device class
MFS	IMS/MFS device type
CSP	CSP/AD device name
DEP	Depth in characters
WID	Width in characters
DPP	Depth in pels
WIP	Width in pels
PDB	Maximum partitions buffer size
PSB	Maximum scroll buffer size
CVM	Character cell size vertical, minimum
CVX	Character cell size vertical, maximum
CHM	Character cell size horizontal, minimum
CHX	Character cell size horizontal, maximum
PFK	Number of PF keys
NPA	Number of partitions
PGH	Number of horizontal gaps between partitions
IBM	An **I** indicates that this device entry is provided by IBM
EXS	An **E** indicates an extendable device size
INP	An **I** indicates that this is an input device
OUT	An **O** indicates that this is an output device
ADJ	An **A** indicates that adjacent fields are possible
NUM	An **N** indicates the numeric lock feature
PEN	A **P** indicates that a light-pen is available
COL	A **C** indicates the color feature
HIL	An **H** indicates the extended highlighting feature
PSS	A **P** indicates the programmed symbol set feature
DBC	A **D** indicates DBCS support
VAL	A **V** indicates the field validation feature
LIN	An **L** indicates the field outlining feature
MIX	An **M** indicates mixed EBCDIC/DBCS
TRA	A **T** indicates the transparency feature
CAT	A **C** indicates character attributes
TAB	A **T** indicates tabulators
LDC	An **L** indicates a logical device code
OBF	An **O** indicates outboard formatting

```
                                                                                          IEIOANPCHPDVLMTCTLO
                                                                                          BXNUDUEOISBAIIRAADB
NAM     DVL SDF BMS GDD MFS     CSP      DEP WID DPP WIP PDB PSB CVMCVXCHMCHXPFKNPAPGHPGV MSPTJMNLLSCLNXATBCF
----------------------------------------------------------------------------------------------------------------
3270        F3          3270              24  80                              24          I IO   CHPDVLM
3270-1      F1  L       3270-A5           12  40                              24          I IO
3270-2      F2  M       3270-A2           24  80                              24          I IO
ALL         F4          3270-A2          240 240                              24          I IO   CHPDVLM
CRLP        F5  A                         12  80                                          I
TAPE        F6  B                         12  80                                          I
DISK        F7  C                         12  80                                          I
TWX         F8  D                         12  80                                          I
1050        F9  E                         12  80                                          I
2740        C1  F                         12  80                                          I
2741        C2  G                         12  80                                          I
2770        C3  I                         12  80                                          I
2780        C4  J                         12  80                                          I
3780        C5  K                         12  80                                          I
INTLU       C6  P                         12  80                                          I               TL
3767        C7  P   R1          3767      66 132                                          IE              TL
3770I       C8  P                         12  80                                          I               TL
SCS         C9  P                         24  80                                          I               TL
6670        D1  P                         50  80                                          I               TL
2980        D2  Q                         12  40                                          I
2980-4      D3  R                         12  40                                          I
3601        D4  U                          1  40                                          I                L
3653        D5  V                          6  30                                          I
3650UP      D6  W                          3  80                                          I
3650/32     D7  X                         23  80                                          I                 O
BCHLU       D8  Y                         12  80                                          I               TL
3770B       D9  Y                         12  80                                          I               TL
274X                    274X              55 132                                          I IOA
3270,1                  3270,1            12  40                              24          I IO   CHP V
3270,2                  3270,2            24  80                              24          I IO   CHPDVLM
3270-A1                 3270-A1           12  80                              24          I IO   CHP V
3270-A2                 3270-A2           24  80                              24          I IO   CHPDVLM
3270-A3                 3270-A3           32  80                              24          I IO   CHP V
3270-A4                 3270-A4           43  80                              24          I IO   CHP V
3270-A5                 3270-A5           12  40                              24          I IO   CHP V
3270-A6                 3270-A6            6  40                              24          I IO   CHP V
3270-A7                 3270-A7           27 132                              24          I IO   CHP V
3270-A8                 3270-A8           62 160 751 960          12 31  6 12 24 16       IEIO   CHP V
3270P                   3270P             55 132                                          I  OA
3270P,1                 3270P,1           55 132                                          I  OA
3270P,2                 3270P,2           55 132                                          I  OA
FIN                     FIN               24  80                                          IEI A
FIDS                    FIDS               6  40                                          I  OA
FIDS3                   FIDS3             12  40                                          I  OA
FIDS4                   FIDS4             16  64                                          I  OA
FIDS7                   FIDS7             24  80                                          I  OA
FIJP                    FIJP              55  80                                          I  OA
FIPB                    FIPB              55 100                                          I  OA
FIFP                    FIFP              55  80                                          I  OA
FIFP132                 FIFP              55 132                                          I  OA
3600                    3600              24  80                                          IEI A
36DS                    36DS               6  40                                          I  OA
36DS3                   36DS3             12  40                                          I  OA
36DS4                   36DS4             16  64                                          I  OA
36DS7                   36DS7             24  80                                          I  OA
36JP                    36JP              55  80                                          I  OA
36PB                    36PB              55 100                                          I  OA
36FP                    36FP              55  80                                          I  OA
36FP132                 36FP              55 132                                          I  OA
SCS1                    SCS1              55 132                                          I IOA     D LM
SCS2                    SCS2              55  80                                          I IOA
```

Figure 54 (Part 1 of 3). The device table

```
                                                                                          IEIOANPCHPDVLMTCTLO
                                                                                          BXNUDUEOISBAIIRAADB
NAM     DVL SDF BMS GDD MFS     CSP      DEP WID DPP WIP PDB PSB CVMCVXCHMCHXPFKNPAPGHPGV MSPTJMNLLSCLNXATBCF
---------------------------------------------------------------------------------------------------------------
DPM-A1                  DPM-A1                                                            I IO
DPM-A2                  DPM-A2                                                            I IO
DPM-A3                  DPM-A3                                                            I IO
DPM-A4                  DPM-A4                                                            I IO
DPM-A5                  DPM-A5                                                            I IO
DPM-A6                  DPM-A6                                                            I IO
DPM-A7                  DPM-A7                                                            I IO
DPM-A8                  DPM-A8                                                            I IO
DPM-A9                  DPM-A9                                                            I IO
DPM-A10                 DPM-A10                                                           I IO
DPM-A11                 DPM-A11                                                           I IO
DPM-A12                 DPM-A12                                                           I IO
DPM-A13                 DPM-A13                                                           I IO
DPM-A14                 DPM-A14                                                           I IO
DPM-A15                 DPM-A15                                                           I IO
DPM-B1                  DPM-B1                                                            I IO
DPM-B2                  DPM-B2                                                            I IO
DPM-B3                  DPM-B3                                                            I IO
DPM-B4                  DPM-B4                                                            I IO
DPM-B5                  DPM-B5                                                            I IO
DPM-B6                  DPM-B6                                                            I IO
DPM-B7                  DPM-B7                                                            I IO
DPM-B8                  DPM-B8                                                            I IO
DPM-B9                  DPM-B9                                                            I IO
DPM-B10                 DPM-B10                                                           I IO
DPM-B11                 DPM-B11                                                           I IO
DPM-B12                 DPM-B12                                                           I IO
DPM-B13                 DPM-B13                                                           I IO
DPM-B14                 DPM-B14                                                           I IO
DPM-B15                 DPM-B15                                                           I IO
8775-1C             C4          8775-1C   12  80         4004779              24  8  2    IEIO   CHP V
8775-1D             D4          8775-1D   12  80         4004779              24  8  2    IEIO   CHP V
8775-2C             C5          8775-2C   24  80         4004779              24  8  2    IEIO   CHP V
8775-2D             D5          8775-2D   24  80         4004779              24  8  2    IEIO   CHP V
8775-3C             C6          8775-3C   32  80         4004779              24  8  2    IEIO   CHP V
8775-3D             D6          8775-3D   32  80         4004779              24  8  2    IEIO   CHP V
8775-4C             C7          8775-4C   43  80         4004779              24  8  2    IEIO   CHP V
8775-4D             D7          8775-4D   43  80         4004779              24  8  2    IEIO   CHP V
3277-1              A2          3277-1    12  40                              24          I IO
3643-2              A1          3643-2     6  40                                          I IO   CHP V
3643-4              A3          3643-4    16  64                                          I IO   CHP V
ANY-1D              D4          ANY-1D    12  80         4004779              24  8  2    IEIO   CHP V
ANY-2D              D5          ANY-2D    24  80         4004779              24  8  2    IEIO   CHP V
ANY-3D              D6          ANY-3D    32  80         4004779              24  8  2    IEIO   CHP V
ANY-4D              D7          ANY-4D    43  80         4004779              24  8  2    IEIO   CHP V
ANY-5D              D8          ANY-5D    27 132         4004779              24  8  2    IEIO   CHP V
ANY-D               D9          ANY-D     62 160 751 960          12 31  6 12 24 16       IEIO    HP
PRINTER             P1          PRINTER   66 132                                          IE O
PRINT-B             Q1          PRINT-B   66 132                                          IE O
5550D               K5          5550D     24  80                              24          I IO   CH D LM
A1                  A1          3643-2     6  40                              24          I
A2                  A2          3277-1    12  40                              24          I
A3                  A3          3643-4    16  64                              24          I
A4                  A4          3278-1    12  80                              24          I
A5                  A5          3278-2    24  80                              24          I
A6                  A6          3278-3    32  80                              24          I
A7                  A7          3278-4    43  80                              24          I
A8                  A8          3278-5    27 132                              24          I
B4                  B4          3278-1B   12  80                              24          I IO   CHP V
B5                  B5          3278-2B   24  80                              24          I IO   CHP V
B6                  B6          3278-3B   32  80                              24          I IO   CHP V
B7                  B7          3278-4B   43  80                              24          I IO   CHP V
B8                  B8          3278-5B   27 132                              24          I IO   CHP V
```

Figure 54 (Part 2 of 3). The device table

```
                                                                                          IEIOANPCHPDVLMTCTLO
                                                                                          BXNUDUEOISBAIIRAADB
NAM     DVL SDF BMS GDD MFS     CSP      DEP WID DPP WIP PDB PSB CVMCVXCHMCHXPFKNPAPGHPGV MSPTJMNLLSCLNXATBCF
---------------------------------------------------------------------------------------------------------------
C4                  C4          8775-1C   12  80                              24          I      CHP V
C5                  C5          8775-2C   24  80                              24          I      CHP V
C6                  C6          8775-3C   32  80                              24          I      CHP V
C7                  C7          8775-4C   43  80                              24          I      CHP V
D0                  D0                     6  20                              24          I      CHP V
D1                  D1                     6  40                              24          I      CHP V
D2                  D2                    12  40                              24          I      CHP V
D3                  D3                    16  64                              24          I      CHP V
D4                  D4          8775-1D   12  80                              24          I      CHP V
D5                  D5          8775-2D   24  80                              24          I      CHP V
D6                  D6          8775-3D   32  80                              24          I      CHP V
D7                  D7          8775-4D   43  80                              24          I      CHP V
D8                  D8                    27 132                              24          I      CHP V
D9                  D9          ANY-D     62 160                              24          I      CHP V
K5                  K5          5550D     24  80                              24          I      CH D LM
K6                  K6                    32  80                              24          I IO   CH D LM
K7                  K7                    43  80                              24          I IO   CH D LM
P1                  P1          PRINTER   66 132                                          I  O
Q1                  Q1          PRINT-B   66 132                                          I  O
R1                  R1          3767      66 132                              24          I IO            TL
V1                  V1          5550P     66 158                                          I IO      D LM
ISPFDEV             D5  3270-A2 3278-2B   24  80                              24          IEIO   CH D LM
3278-1      01  1   A4  3270-A1 3278-1    12  80                              24          I IO   CHP V
3278-2      02  2   A5  3270-A2 3278-2    24  80                              24          I IO   CHP V
3278-3      03  3   A6  3270-A3 3278-3    32  80                              24          I IO   CHP V
3278-4      04  4   A7  3270-A4 3278-4    43  80                              24          I IO   CHP V
3278-5      05  5   A8  3270-A7 3278-5    27 132                              24          I IO   CHP V
3278-1B     06  1   B4  3270-A1 3278-1B   12  80                              24          I IO   CHP V
3278-2B     07  2   B5  3270-A2 3278-2B   24  80                              24          I IO   CHP V
3278-3B     08  3   B6  3270-A3 3278-3B   32  80                              24          I IO   CHP V
3278-4B     09  4   B7  3270-A4 3278-4B   43  80                              24          I IO   CHP V
3278-5B     0A  5   B8  3270-A7 3278-5B   27 132                              24          I IO   CHP V
3279-2B     17  2   D5  3270-A2 3278-2B   24  80                              24          I IO   CHP V
3279-3B     18  3   D6  3270-A3 3278-3B   32  80                              24          I IO   CHP V
3290        0D  9       3270-A8           62 160 751 960          12 31  6 12 24 16       IEIO    HP
8775        0C  7       3270-A4           43  80         4004779              24  8  2    IEIO   CHP V
5550        1C  8   K5  3270-A2 5550D     24  80                              24          I IO   CH D LM
3278-52     0B  8   K5  3270-A2 5550D     24  80                              24          I IO      D
3278-K1     1B  8       3270-A2           24  80                              24          I IO   CH D LM
5550P       2C  0   V1  SCS1    5550P    127 158                                          IE OA     D LM
3283-52             V1          5550P    127 158                                          IE O      D
3283-K1     2B  0                        127 158                                          I  O      D LM
3179            2       3270-A2           24  80                              24          I IO   CHP V
3179-G          3       3270-A3           32  80                              24          I IO   CHP V
3180            5       3270-A2           27 132            7680              24  1       IEIO   CHP V
```

Figure 54 (Part 3 of 3). The device table

Appendix B. Servicing SDF II

Service to SDF II is provided by means of a program temporary fix (PTF), which you install using MSHP.

If a PTF refreshes any parts of SDF II that have been customized in your installation, you may need to apply those modifications again. If a PTF replaces members with type A, X, or Z, you may need to take additional action, depending on the member type:

Assembler source files

These are shipped and serviced with member type A. If a member that you have customized is affected by service, you must re-apply your modifications.

DGIJOBFS.A

Must be customized as described under "Customizing the full-screen start-up job" on page 10.

DGIYTUC.A, DGIYTLC.A, and DGIYTDC.A

Must be customized as described under "Changing the translation tables" on page 15.

Panels, tables, and skeletons

SDF II panels, tables, and skeletons are type A members, which may be affected by customization. Check your SDF II customization library for an object with a matching name. If you find one, rerun the corresponding customization dialog.

The following members may be affected by customization:

DGIKF*.A
DGIUS*.A
DGIYS*.A
DGIYY*.A

For more information, see Chapter 4, "Customizing SDF II profiles and tables" on page 24.

REXX procedures with type X

In a PTF, type X members are REXX procedures that are not subject to customization. After installing a PTF, rename any type X member to type PROC.

Members with type Z

Type Z members are REXX and JCL procedures, and samples, which are subject to customization. If a PTF replaces a type Z member that has been adapted for your installation, re-apply your modifications. If it is a procedure, change its type to PROC.

Appendix C. Diagnosing SDF II problems

This appendix summarizes the steps you can take when a problem occurs.

Preparing a problem record

Search the IBM Software Support Databases for known problems, using the symptom as a keyword. If the keyword search is unsuccessful or if you do not have access to the databases, contact your IBM Support Center. You should be prepared to supply the following information:

- Your operating system and release level
- Release and PTF level of SDF II
- The keyword set that was used to search the software support database
- A free-form problem description, including all symptoms

Gathering documentation for the problem record

You will be asked for information that describes the failure in SDF II. You may be asked to include some items from the following list:

- A failing test case, made as short as possible while still demonstrating the failure. If it is too long to allow easy keyboard entry from your written description, it should be on magnetic tape (unlabeled or standard labeled tape).

  It is best to include the actual SDF II object that showed the failure.

- A hard copy of job control statements.
- An example of the failing output and of the expected output.
- Output from the SDF II trace facility (SYSLST).
- Current service level. The consolidated software inventory, such as a library list.
- A storage dump.
- Console output from the system on which the job was run.
- For an import utility failure:
  - The input to the failing utility
  - The imported objects in SDF II that fail or that show the problem

With this information, IBM Support Center personnel can try to reproduce the failure and observe its symptoms.

Submitting printed material as problem documentation

Ensure that any printed material, including dumps, is clearly marked with the assigned problem number in the top right-hand corner.

The SDF II trace facility

This section explains how to start and use the SDF II trace facility and how to interpret the trace listing.

The SDF II trace facility produces a chronological listing of SDF II services and indicates where in SDF II each event occurred. The listing also shows whether the event was successful.

Using the trace facility may affect the performance of SDF II, because all the trace information is written to SYSLST.

Starting the SDF II trace facility in an online environment

To start the SDF II trace facility in an online environment, enter on an SDF II panel:

trace on Starts the trace for SDF II services.
trace all Starts the trace for both SDF II and SDF2/DM services. It is required only for problems in the SDF2/DM services.

To stop the SDF II trace facility, enter:

trace off

Running the SDF II trace facility in a batch environment

To control the SDF II trace facility in batch, add the trace option to the REQU= parameter in the SDF II start-up job. For example:

```
// EXEC PROC=DGIPROC, REQU='/ON'
```

The trace options are:

{**/ON** | **/ALL**}

These options have the following meanings:

/ON Starts the trace for SDF II services.
/ALL Starts the trace for both SDF II and SDF2/DM services. It is required only for problems in the SDF2/DM services.

The trace listing

A trace entry is written whenever an event starts or stops. It appears as:

```
hh:mm:ss.ttt  nnnnnnnn oooo mmmmmmmm xrrrr parameters
```

This is what the entries mean:

hh:*mm*:*ss*.*ttt* The time stamp.

nnnnnnnn The name of the module that issued the request.

oooo The hexadecimal offset of the instruction in the module that issued the request.

mmmmmmmm The name of the requested service or the name of additional information.

x Indicates whether the trace entry is produced before (>) or after (<) the service is performed.

rrrr The return code (decimal) in register 15.

A zero (0) in this position means that the service or function completed normally.

parameters A string of parameters that might include the address of the parameter list and the processing time taken.

Abend codes

After an abnormal end, SDF II normally displays panel DGIYE80. This panel contains:

- Abend code
- Reason code
- Offset and SDF2/DM error message number
- Short message
- Long message

If panel DGIYE80 is not displayed, the abend was issued by SDF2/DM or by the operating system.

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

abend. Abnormal end of task.

action bar. In Common User Access architecture, the area at the top of a window that contains choices that give a user access to actions available in that window.

action bar choice. A textual item on an action bar, which provides access to menus that contain choices that can be applied to an object.

active partition. The partition that contains the cursor. It can be scrolled vertically. While a partition is active, the cursor “wraps around” at the viewport boundaries, and the *ENTER* key (or input key) transmits data from that partition only.

adjunct. An optional field in the data structure that is added to a field in the data structure, which contains data to be displayed. It enables the application program to vary a specific presentation attribute (or set of attributes) at run time.

AID. See attention identifier.

AID table. A table that assigns values to actions performed by the user of the application program. An action may be, for example, the pressing of a program function key. The values are used by the application program.

APAR. Authorized program analysis report.

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 attribute. A property of a variable field, such as justification of data in the data structure. Contrast with presentation attribute.

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

application element. Any single item in the data structure.

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

area. A rectangular part of a format, whose contents (text or graphics) are provided at run time by the application program. See graphic area and dynamic area.

area attribute. An attribute that affects the properties of an area. It can be, for example, extendable or scrollable.

area mark. A mark used to define an area (see area), such as a graphics area or a dynamic area.

array. A named, ordered collection of variable fields, all of which have identical names and attributes. An array has a specified occurrence number denoting the number of elements in the array. See horizontal array and vertical array.

array index. A number in parentheses that appears next to the name of an array. For example, in the name of the element *a(3)* of the array *a*, *3* is the array index.

assembler (ASM). A computer program that converts assembly language instructions into object code.

attention identifier (AID). A character in a data stream indicating that the user has pressed a key, such as the Enter key, that requests an action by the system.

attribute. See presentation attribute and application attribute. See application attribute, area attribute, background attribute, character attribute, field attribute, inherent attribute, and presentation attribute.

attribute descriptor. A symbol that denotes a set of attributes.

attribute line. A line showing the attribute descriptors assigned to the field.

autosave. An automatic save facility in which the user can define a specific number of alterations after which a temporary save occurs automatically.

autosave library. A library in which the saved objects are stored.

B

background attribute. The attributes associated with background text.

background text. All text on a panel that is not within a constant or variable field.

base name. The name that is used in a based data structure as a pointer variable that identifies the location of the data.

block. In SDF II, a rectangular part of a format that is defined by the position command for such commands as moveblock or delblock.

C

C. A high-level programming language.

character attribute. An attribute that applies to a single character.

CICS/BMS. Customer Information Control System/Basic Mapping Support.

COBOL. A high-level programming language, based on English, that is used primarily for business applications.

Common User Access (CUA) architecture. Guidelines for the dialog between a person and a workstation or terminal.

constant field. In SDF II, a field that contains constant text, which has attributes that differ from background attributes. Contrast with variable field.

control table. (1) In IMS/MFS, a user-defined table of operator control functions; a specific control function is invoked when the input device data or data length satisfies a predefined condition. (2) In SDF II, an object that corresponds to an operator control table in IMS/MFS.

conversion. A process by which an object defined for a specific target system is changed so that it becomes an object for another target system. The converted object will retain those properties which are supported by the new target system.

Cross System Product (CSP/AD and CSP/AE). A set of licensed programs designed to permit the user to develop and run applications using independently defined maps (display and printer formats), data items (records, working storage, files, and single items), and processes (logic). The Cross System Product set consists of two parts: Cross System Product/Application Development (CSP/AD) and Cross System Product/Application Execution (CSP/AE).

CSP/AD. Cross System Product/Application Development.

CSP/AE. Cross System Product/Application Execution.

CUA. See Common User Access architecture.

CUA attribute. Synonym for CUA panel element attribute.

CUA panel element. The smallest named part of a panel, such as a title, which is based on CUA architecture.

CUA panel element attribute. In SDF II, any attribute associated with a CUA panel element type. Synonymous with CUA attribute.

CUA panel element type. In SDF II, used as a reference to a class of CUA panel elements. Synonymous with CUA type.

CUA type. Synonym for CUA panel element type.

Customer Information Control System (CICS). An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

D

DASD. Direct access storage device.

data mark. Synonymous with DATAIN/DATAOUT attribute characters in ISPF.

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.

device list. A list of compatible device types. It is defined by the system programmer.

device table. Synonym for device type table.

device type. In SDF II, the name of a device or of a device list.

device type editor. An editor used for creating and maintaining the device type table.

device type table. A table containing the names of all device types supported by SDF II, together with the features available on the devices. It is maintained by the SDF II administrator.

DFLD. Device field.

dialog. (1) The interaction between a user and a computer. (2) In SDF II, one or more panels and associated logic that establish an interactive session between SDF II and a user. A dialog prompts the user to enter information appropriate to the function requested and displays the results.

direct access storage device (DASD). A device in which access time is effectively independent of the location of the data. (A)

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.

DSECT. Dummy control section.

dummy control section (DSECT). A control section that an assembler can use to format an area of storage without producing any object code. (A)

dynamic area. In SDF II, an area that is filled with text at run time by the application program.

E

EBCDIC. See extended binary-coded decimal interchange code.

emphasis class. In SDF II, a set of predefined attributes. Emphasis classes can be specified for fields, marks, and attribute descriptors.

EXEC. An executable procedure that contains operating system commands and execution control statements.

extended attribute. Any one of the color, highlight, programmed symbol set, outlining, mixed, or validation attributes.

extended binary-coded decimal interchange code (EBCDIC). A coded character set of 256 8-bit characters.

extended external source format. In SDF II, an extension of CSP/AD's external source format representing certain properties of CICS/BMS and IMS/MFS. See external source format.

external source format. CSP/AD's external source format is a commonly used means of representing applications and panels in an AD/Cycle framework. The format consists of a readable syntax of mark-up tags and attributes.

F

field attribute. A defined characteristic of a field, such as protected or unprotected, alphanumeric or numeric, detectable or nondetectable, displayable or nondisplayable, or intensity. See presentation attribute and inherent attribute.

field format. A field property that determines the character set that can go into a given field.

format. A format is part of a panel. It 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 on a screen by a user. A format may consist of different definitions for different device types. These definitions are called format instances.

format element. A part of a format, such as a variable field, a constant field, a dynamic area, a graphic area, a repeat format, or an include panel.

format instance. A part of a format that defines the appearance of data for a particular device type.

format mode. One of the four modes in which SDF II can display the layout of a panel. In this mode, marks show the extent of fields and areas. Contrast with initial value mode, name mode, and sample value mode.

G

GDDM–IMD. Graphical Data Display Manager — Interactive Map Definition.

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

graphical data display manager (GDDM). A group of routines that allows pictures to be defined and displayed procedurally through function routines that correspond to graphic primitives.

H

horizontal array. An array that is read from left to right and line by line. For example:

choice (1)	choice (2)
choice (3)	choice (4)

See array and vertical array.

I

import. In SDF II, a process by which objects are imported into SDF II from one of the supported target systems, from SDF/CICS, or from an external source format structure.

IMS/MFS. Information Management System/Message Format Service.

include panel. A panel that is included in another panel. Examples are headers and trailers.

Information Management System/Virtual Storage (IMS/VS). A database/data communication (DB/DC) system that can manage complex databases and networks. Synonymous with IMS.

inherent attribute. An attribute that can be defined for variable and constant field marks, and data marks. After the field is defined, inherent attributes cannot be changed.

initial mode. In SDF II, one of the four modes in which SDF II can display the layout of a panel. In this mode, the Format window shows each initial value in its variable field. Contrast with format mode, name mode, and sample mode.

initial value. A value the SDF II user assigns to a variable field. The application program displays this value at run time if no value has been provided by the application.

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. (A)

ISPF. Interactive System Productivity Facility.

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

M

mark. In SDF II, a character used to define a format element, such as a field or area, or to provide some editing function. Examples include area marks, character marks, separator marks, and spacer marks.

MRI. Machine-readable information.

machine-readable information (MRI). All textual information contained in a program, such as a system control program, an application program, or microcode. MRI includes all information that is presented to or received from a user interacting with a system. This includes menus, prompts, messages, report headings, commands, and responses. MRI may appear on printers or on display panels. (A)

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

N

name mode. One of the four modes in which SDF II can display the layout of a panel. In this mode, the Format window shows the name of the variable field in the field. Contrast with format mode, initial mode, and sample mode.

national language support (NLS). The modification or conversion of a United States English product to conform to the requirements of another language or country. This can include the enabling or retrofitting of a product and the translation of nomenclature, MRI, or documentation of a product. (A)

NLS. National language support.

nonprogrammable terminal (NPT). In Basic Common User Access architecture, a terminal attached to a host processor in which most of the user-interface functions are controlled by the host processor. (A)

NPT. Nonprogrammable terminal.

O

object. In SDF II a panel, panel group, partition set, or AID table stored in an SDF II library.

Figure 55 shows the equivalents for these objects in the target systems.

Figure 55. SDF II objects and target system equivalents

SDF II Object	IMS/MFS	CICS/BMS [1]	GDDM–IMD	CSP/AD or VisualGen	ISPF
Panel	Format set	Map	Map	Map	Panel
Panel group		Map set	Map group	Map group	
Partition set	Partition definition block	Partition set			
AID table	PF key parameter of the DEV statement		AID table		
Control table	Operator control table				

[1] SDF/CICS uses the same terms as CICS/BMS

operational prototype. A simulation of an application program to test or review simple functions, such as simple database access, scrolling, error reporting, and online help panels. For some application programs, an operational prototype may include some characteristics of a database, including some program code or SDF II dialog manager tables. The operational prototype is used to determine the needs of the user of the application program and to ensure that the application program meets those needs. See simulative prototype.

P

page. In SDF II, part of a format instance that corresponds to an IMS/MFS physical page.

panel. (1) The information that is displayed at any one time on the screen. (2) An SDF II object. It 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 element. (1) An element of a panel as displayed in the Define Panel Instances dialog, which denotes one of the following:

- Format
- Format instance
- Page
- Data structure

(2) A line in the Specify Panel Elements dialog. It is used by the panel construction utility to create one or more fields, panel text, or a repeat format on the panel to be constructed.

panel group. An object within SDF II that contains a list of panel names and describes the properties of these panels.

panel group instance. A part of a panel group that describes the properties of the panel group for a particular device type.

partition. All or a portion of the screen. Data is presented within the partition through a viewport, which is defined when the partition is created.

partition set. An SDF II object that consists of a group of partitions designed to share the same screen.

partition set instance. A part of a partition set that describes the properties of the partition set for a particular device type.

PL/I. A programming language that is designed for use in a wide range of commercial and scientific computer applications. (A)

presentation attribute. An attribute that defines how information is presented on the screen, such as highlighting and color. Contrast with application attribute.

program temporary fix (PTF). A temporary solution or bypass of a problem diagnosed by IBM as resulting from a defect in a current unaltered release of the program.

prototype. See simulative prototype and operational prototype.

PTF. Program temporary fix.

pull-down. In Common User Access architecture, a list of choices associated with a choice on the action bar. A user selects a choice from the action bar and a pull-down menu appears.

pull-down choice. A textual item on a menu. A user selects a choice to work with an object in some way.

R

reference name. A 1- or 2-character name used by SDF II as a synonym for the name of a variable field.

repeat format. A rectangular part of the format that can be repeated down a panel. All instances of a repeat format must have the same variable fields at the same relative horizontal positions as in the source format.

Report Program Generator II (RPG II). A commercially oriented programming language specifically designed for writing application programs intended for business data processing. (A)

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

REXX. Restructured Extended Executor language.

RPG II. Report Program Generator II.

S

sample mode. One of the four modes in which SDF II can display the layout of a panel. In this mode, the Format window shows each sample value in its variable field. Contrast with format mode, initial mode, and name mode.

sample value. A value the SDF II user assigns to a variable field. SDF II displays this value when the panel is tested or during prototype simulation.

Screen Definition Facility/Customer Information Control System (SDF/CICS). An online application development tool used by application programmers to define or edit maps, map sets, and partition sets for CICS/VS Basic Mapping Support. (A)

Screen Definition Facility II (SDF II). An interactive application development tool that helps application developers to define, maintain, import, and generate screen objects, such as panels, panel groups, partition sets, attention identifier (AID) tables, and control tables, as appropriate, for its target systems.

scrollable area. The window in the main panel behind which a scrollable area format can be scrolled.

scrollable area format. A separate format used with a scrollable area.

SDF/CICS. Screen Definition Facility/Customer Information Control System.

SDF II. Screen Definition Facility II.

SDF II dialog manager. The dialog management component of Screen Definition Facility II.

SDF2/DM. SDF II dialog manager.

separator. In SDF II, a mark used to separate the length of a field, its name, and its mark.

shift-in character (SI). A code extension character used to terminate a sequence that has been introduced by the shift-out character to make effective the graphic characters of the standard character set. (I) Contrast with shift-out character.

shift-out character (SO). In SDF II, a code extension character that substitutes, for the graphic characters of the standard character set, DBCS. Contrast with shift-in character.

SI. Shift-in character.

simulative prototype. A simulation of a series of panels used by an application program 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.

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

skip after attribute. In SDF II, a presentation attribute that causes the cursor to skip to the next unprotected field when the field in which the cursor is located has been filled.

SO. Shift-out character.

spacer. In SDF II, a mark that positions information on lines during panel definition; it is typically used for centering.

specification object. Synonym for object.

T

target system. A system under which the application using an SDF II panel can be run. For example, CICS/BMS, CSP/AD, VisualGen, ISPF, GDDM–IMD, and IMS/MFS.

U

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

V

variable field. A field in which data may be changed by the application program or by the user. It has a character string, which can be empty, defined at run time as its contents. If no contents are provided at run time, the initial value, if defined at specification time, is taken as default instead. Contrast with constant field.

vertical array. An array that is read from top to bottom and column by column. For example:

choice (1)	choice (3)
choice (2)	choice (4)

See array and horizontal array.

Virtual storage extended (VSE). An IBM licensed program whose full name is the Virtual Storage Extended/Advanced Function. It is a software operating system controlling the execution of programs. (A)

VSE. Virtual storage extended.

W

window. In SDF II, a rectangular part of the screen where scrollable data is displayed and can be manipulated.

SDF II publications

The SDF II Release 6 publications are:

SDF II Licensed Program Specifications, GH12-6318
Contains the product specifications and warranty information.
Audience: Data processing manager, system programmer.

Introducing SDF II Release 6 for VSE, GH12-6314
Summarizes the functions, uses, requirements, and advantages of SDF II.
Audience: Data processing manager, system programmer.

SDF II General Introduction, SH12-6315
Introduces SDF II to new users and explains how to define simple panels. It also explains how to prototype the flow of panels and main functions of an application.
Audience: System programmer, application programmer, application user.

SDF II Primer for CICS/BMS Programs, SH12-6313
Explains how to use SDF II to develop objects for applications that run under CICS/BMS.
Audience: System programmer, application programmer, application user.

SDF II Run-Time Services, SH12-6312
Provides a comprehensive reference to the language and functions of the SDF II dialog manager (SDF2/DM).
Audience: System programmer, application programmer.

SDF II Administrator's Guide, SH12-6311
Describes how to customize SDF II on a VSE system. It also explains how to import objects into SDF II, how to set up and work with libraries, how to run SDF II from batch, and how to identify and report problems in SDF II to IBM support personnel.
Audience: System programmer, application programmer.

SDF II Reference Summary, SX12-5012
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, application user.

Related reading

CICS Application Programming Guide, SC33-0712
CICS Application Programmer's Reference, SC33-0711
CICS System Programmer's Reference, SC33-0713
CSP/AD External Source Format Reference, SH20-6433
ISPF/PDF for VM/SP Services, SC34-4012
OS/390 ISPF Services Guide, SC28-1272
REXX/VSE Reference Guide, SC33-6642
SDF/CICS Release 5 Program Reference, SH19-8105
VisualGen External Source Format Reference, SH23-6585
VisualGen GUI User's Guide and Reference, SH23-6559
VisualGen Planning, SH23-6553
VSE/ESA Administration, SC33-6605
VSE/ESA Planning, SC33-6603
VSE/ESA System Control Statements, SC33-6613
VSE/POWER Administration and Operation, SC33-6633

Index

Your comments, please ...

Screen Definition Facility II for VSE
Administrator's Guide
Release 6

Publication No. SH12-6311-00

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.

If you mail this form to us, be sure to print your name and address below if you would like a reply.

You can also send us your comments using:

- A fax machine. The number is: +43–1–21145–4490
- Internet. The address is: kpesen@vnet.ibm.com.

  Please include the title and publication number (as shown above) in your reply.

Name

Company or Organization

Phone No.

Address

Your comments, please ...
SH12-6311-00

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM Vienna Software
SDF Development/Service
Department 00/174
Lassallestrasse 1
A–1020 Vienna
Austria
Europe

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Program Number: 5746-XXT

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SH12-6311-00