

# Getting Started With IOF

DPS7000/XTA  
NOVASCAL 7000

Job Control and IOF



REFERENCE  
47 A2 35UJ 00



# DPS7000/XTA NOVASCALÉ 7000 Getting Started With IOF

Job Control and IOF

January 1995

BULL CEDOC  
357 AVENUE PATTON  
B.P.20845  
49008 ANGERS CEDEX 01  
FRANCE

REFERENCE  
47 A2 35UJ 00

The following copyright notice protects this book under Copyright laws which prohibit such actions as, but not limited to, copying, distributing, modifying, and making derivative works.

Copyright © Bull SAS 1995

Printed in France

Suggestions and criticisms concerning the form, content, and presentation of this book are invited. A form is provided at the end of this book for this purpose.

To order additional copies of this book or other Bull Technical Publications, you are invited to use the Ordering Form also provided at the end of this book.

### **Trademarks and Acknowledgements**

We acknowledge the right of proprietors of trademarks mentioned in this book.

Intel® and Itanium® are registered trademarks of Intel Corporation.

Windows® and Microsoft® software are registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark in the United States of America and other countries licensed exclusively through the Open Group.

Linux® is a registered trademark of Linus Torvalds.

*The information in this document is subject to change without notice. Bull will not be liable for errors contained herein, or for incidental or consequential damages in connection with the use of this material.*

## Preface

IOF is the GCOS 7 Interactive Operating Facility (IOF).

This manual provides basic information about how to get started with IOF.

### Audience

This manual is principally directed at IOF novices who are unsophisticated, or occasional end-users. However, this manual can be useful to the more sophisticated user who is not familiar with IOF, and there is some information included for system managers who fall into that category.

### Document structure

Chapter 1 introduces IOF and its facilities, and discusses a number of key concepts and other basic information about user access, as well as illustrating where IOF fits into a DPS installation.

Chapter 2 covers how to connect to IOF, describing how to log on and off, how to use the menus and get help, and gives an example IOF session.

Chapter 3 discusses some more sophisticated operations, including library selection, creation and deletion, printing, and tutorials for the Full Screen Editor (FSE), and using GCL (GCOS 7 Command Language) procedures.

Chapter 4 supplies some further information about creating libraries and catalogs and cataloged files.

## Getting Started With IOF

### Associated documents

Bull customers can order any of the manuals listed below from the address given on the title page of this manual.

The following publications may be of interest to IOF users.

<i>Catalog Management User's Guide</i> .....	47 A2 35UF
<i>Full Screen Editor User's Guide</i> .....	47A2 06UP
<i>IOF Terminal User's Reference Manual Part 1</i> .....	47 A2 38UJ
<i>IOF Terminal User's Reference Manual Part 2</i> .....	47 A2 39UJ
<i>IOF Terminal User's Reference Manual Part 3</i> .....	47 A2 40UJ
<i>Library Maintenance User's Guide</i> .....	47 A2 02UP
<i>Library Maintenance Reference Manual</i> .....	47 A2 01UP
<i>GCL Programmers Manual</i> .....	47 A2 36UJ
<i>DMU User's Guide</i> .....	47 A2 26UF
<i>Administering the Storage Manager</i> .....	47 A2 36UF

## Table of Contents

<b>1.</b>	<b>Introduction .....</b>	<b>1-1</b>
1.1	WHAT IS IOF? .....	1-1
1.2	KEY CONCEPTS .....	1-3
1.2.1	Libraries and Members .....	1-3
1.2.2	Files, Directories, and Catalogs .....	1-3
1.2.3	Source Text Editors.....	1-7
1.2.4	User Profile.....	1-8
1.2.5	Menus .....	1-8
1.2.6	Help .....	1-8
1.2.7	GCL Commands and Directives.....	1-9
1.3	INFORMATION YOU NEED TO KNOW TO USE IOF ....	1-10
1.3.1	User name .....	1-10
1.3.2	Password.....	1-10
1.3.3	Project.....	1-11
1.3.4	Billing.....	1-11
1.4	IOF IN A DPS 7000 INSTALLATION .....	1-12

Getting Started With IOF

<b>2.</b>	<b>Connecting to IOF .....</b>	<b>2-1</b>
2.1	LOGGING ON .....	2-1
2.1.1	Logging on to the Network Software.....	2-2
2.1.2	Logging on to IOF with Default Options .....	2-4
2.1.3	Logging on to IOF with Specified Options .....	2-6
2.1.4	Resolving Difficulties .....	2-10
2.1.5	Using Startups .....	2-11
2.2	USING MENUS .....	2-14
2.2.1	Executing Commands.....	2-17
2.2.2	Supplying Parameter Values .....	2-19
2.3	GETTING HELP .....	2-21
2.3.1	Help Texts .....	2-21
2.3.2	An Example of Getting Help .....	2-23
2.4	USER PROFILES.....	2-25
2.5	LOGGING OFF .....	2-27
2.6	OUTLINE OF AN IOF SESSION.....	2-28
2.6.1	Logging On .....	2-28
2.6.2	Executing a Command.....	2-30
2.6.3	Logging Off .....	2-30
<b>3.</b>	<b>Developing Applications .....</b>	<b>3-1</b>



Table of Contents

<b>3.1</b>	<b>SELECTING DEFAULT LIBRARIES</b> .....	3-1
<b>3.1.1</b>	<b>Input Libraries</b> .....	3-2
<b>3.1.2</b>	<b>Output Libraries</b> .....	3-3
<b>3.1.3</b>	<b>Output Library Listing</b> .....	3-4
<b>3.2</b>	<b>USING FSE</b> .....	3-5
<b>3.2.1</b>	<b>Calling FSE</b> .....	3-6
<b>3.2.2</b>	<b>Creating a Subfile or Member</b> .....	3-10
<b>3.2.3</b>	<b>Modifying a Subfile or Member</b> .....	3-14
<b>3.3</b>	<b>CREATING AN SL LIBRARY</b> .....	3-23
<b>3.3.1</b>	<b>Allocating Library Space</b> .....	3-26
<b>3.3.2</b>	<b>Creating a Directory</b> .....	3-27
<b>3.3.3</b>	<b>Listing Information about a Library</b> .....	3-28
<b>3.4</b>	<b>PRINTING LIBRARY MEMBERS</b> .....	3-33
<b>3.5</b>	<b>DELETING LIBRARIES</b> .....	3-35
<b>3.6</b>	<b>USING GCL PROCEDURES</b> .....	3-37
<b>3.6.1</b>	<b>Creating a GCL Procedure</b> .....	3-37
<b>3.6.2</b>	<b>Executing a GCL Procedure</b> .....	3-43
<b>4.</b>	<b>More About Files and Libraries</b> .....	4-1
<b>4.1</b>	<b>CREATING LIBRARIES</b> .....	4-1
<b>4.2</b>	<b>CREATING CATALOGED FILES</b> .....	4-4

Getting Started With IOF

<b>4.2.1</b>	<b>Creating a Catalog</b> .....	<b>4-4</b>
<b>4.2.2</b>	<b>Creating a Cataloged File</b> .....	<b>4-9</b>
<b>4.2.3</b>	<b>Accessing a Cataloged File</b> .....	<b>4-12</b>

## Table of Contents

## Illustrations

### Figures

1-1	Example of a Tree Structure .....	1-4
1-2	IOF Connections.....	1-13
2-1	Example of a Log-on Screen .....	2-3
2-2	Example of a Menu Screen .....	2-15
2-3	Executing a Command from a Menu.....	2-18
2-4	Example of Supplying Parameter Values .....	2-20
2-5	Help Screen Layout .....	2-21
2-6	Requesting Help for LIB parameter.....	2-23
2-7	LIB parameter Help Text .....	2-24
2-8	Example Response to a DISPLAY_PROFILE Directive...2-26	
2-9	Example of a Menu.....	2-29
3-1	Library Parameter Screen .....	3-7
3-2	Initial FSE Menu .....	3-8
3-3	FSE Create Screen .....	3-10
3-4	Example of FSE Member Screen.....	3-11
3-5	Example of Entering Text in FSE Member .....	3-12
3-6	Example of Entering Text in FSE .....	3-13
3-7	Modify Member FSE Screen .....	3-15
3-8	Example of Text Substitution in FSE .....	3-16
3-9	FSE Text Substitution Results.....	3-17
3-10	Requesting to Append Text.....	3-18
3-11	Appending Text in FSE.....	3-19
3-12	Results of Added Text in FSE .....	3-20
3-13	Renumbering FSE TextTitle .....	3-21
3-14	FSE Renumbered Text Results.....	3-22
3-15	Information Listed about a Library .....	3-29
3-16	Listing File Space .....	3-30
3-17	Example of Listing a Directory .....	3-32
3-18	DPRINT Screen in Novice Mode .....	3-34
3-19	Deleting a Whole Library .....	3-36
3-20	GCL Menu .....	3-39
3-21	Keyword Parameter Screen.....	3-40
3-22	On Screen View of Creating GCL Procedure .....	3-42
3-23	Example of Executing GCL Procedure.....	3-43

Getting Started With IOF

# 1. Introduction

## 1.1 WHAT IS IOF?

The Interactive Operating Facility - IOF - is a time-sharing facility developed to run under GCOS 7. One of its major advantages is its versatility. It allows a number of users to execute programs at the same time, and allows them to interact with those programs while they are being executed.

IOF can also provide anything that you can request from the system in the following contexts:

- batch
- remote batch
- transactional

IOF allows users to work in IOF in two different modes:

- novice mode
- line mode

In novice mode the user works exclusively with menus and screens. In line mode, users generally work at the prompt, but the user can easily request a menu or screen at any point if necessary.

The novice mode and line mode facilities are another aspect of the IOF operating system's versatility, and contribute to its capacity to support a wide range of potential users who have varied, and sometimes conflicting, needs and objectives.

## Getting Started With IOF

Here are four typical potential IOF user profiles:

- **Unsophisticated or Occasional End-Users**, these are non-programmers who require an easy, pragmatic way to use the system. Their objective is to obtain results as quickly and painlessly as possible.
- **System Operators**, who need to be able to react promptly and effectively to any request from the system. Their objective is production.
- **Systems Engineers and Sophisticated Users**, who need an in-depth knowledge of the system, including its subtleties and limitations. They need to obtain results as quickly and painlessly as possible.
- **System and Installation Managers**, who need to control the use of the system and the service it provides to its users. Their objectives concern decision-making and management.

**Note:** This manual is mainly intended for **Unsophisticated or Occasional End-Users**.

IOF supports all of the above users, as well as others, without over-emphasizing or neglecting any of them.

## Introduction

### 1.2 KEY CONCEPTS

Some concepts useful for getting started with IOF are briefly discussed in this section.

#### 1.2.1 Libraries and Members

Once you have successfully logged onto the system, you need a place where you can store your programs and data so that you can find them again. This place is called a **library**.

A library is a file organized into a series of subfiles called **members**. Each member contains a separate unit of information. These library members make it easy for you to retrieve, edit, and store the various types of information you use in your daily work.

**Note:** In a large project the library is defined for you and you will probably work in a context where you do not need to know the name or location of the library. If you have your own project, one of your first tasks is to define one or more libraries.

#### 1.2.2 Files, Directories, and Catalogs

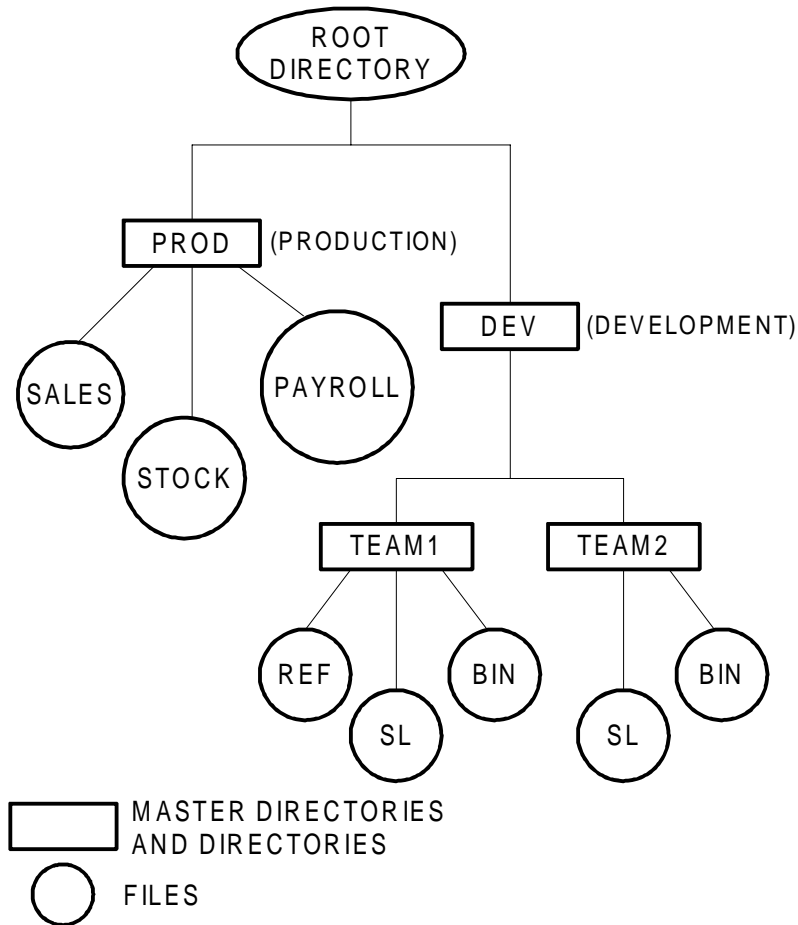
In the GCOS 7 environment, files are organized into groups which are linked together in a tree structure. Each group of files is associated with a directory, and each file in a group is said to be located in that directory.

The directory at the base of the tree is called the root. The first directories in the tree structure are called master directories. Each master directory can have other directories below it.

The links between a master directory and its files are stored in a special file called a catalog.

Getting Started With IOF

The diagram below (Figure 1-1) represents a tree structure with directories and files.



**Figure 1-1. Example of a Tree Structure**



## Introduction

There are 2 types of files: **temporary** and **permanent**.

### Temporary files

A **temporary file** can exist only for the duration of a job or session. In general, a temporary file is created during the execution of a program. It is then passed to another program, used by this program, and finally deleted. Temporary files that are not passed to another program are deleted automatically when the program terminates. Temporary files that are passed on to another program, are automatically deleted at the end of the session.

### Permanent files

A **permanent file** is a file intended to exist for more than a single session. Permanent files can be **cataloged** or **uncataloged**. Once created by a user or program, a permanent file can only be deleted on explicit request.

A cataloged file is one whose name and attributes are stored in another file called a catalog. Most permanent files are cataloged.

A **catalog** is a file which consists of a set of records, each record describing a cataloged object (in this case a file) identified by a unique object name.

Once a file is **cataloged**, you only have to enter the name of the file (the object name) to reference it. The system retrieves from the catalog the information relative to the directory(ies) in which the file resides.

## Getting Started With IOF

**Cataloged** files have certain advantages over uncataloged files: you can keep a number of attributes in a catalog, including, access rights, the authorized level of sharing (for example, concurrent access by several users), the type of file integrity, etc., as well as information such as the last update date, last file version, and the current generation number.

The name and attributes of an **uncataloged** file are not stored in a catalog. Therefore, to reference an **uncataloged** file you must provide both its name and the name of the disk on which it resides.

It is recommended that you catalog your permanent files.

## Introduction

### 1.2.3 Source Text Editors

There are 2 products available for program entry and editing: **FSE** (Full Screen Edit) and **EDIT** (Text Editor). You can activate both of them as stand-alone processors, or from within `MAINTAIN_LIBRARY` (and some other processors) as commands.

#### **FSE**

**FSE** is for editing and entering texts from full-screen terminals. You can refer to lines in the same manner as in **EDIT**. However, full screen facilities are provided so you can enter modifications directly on the screen using the cursor and local editing keys. This makes editing much easier because you can see each modification in the context of lines surrounding it. **FSE** is recommended for novice users.

Refer Section 3.2 of this manual for more information about **FSE**.

#### **EDIT**

**EDIT** is best adapted to serial terminals and batch usage. It was designed before full screen facilities were available. **EDIT** is not discussed in this manual, and not recommended for the users this manual addresses.

## Getting Started With IOF

### 1.2.4 User Profile

Different users have different requirements, depending on their experience, personal preferences, and the type of work they do. It is IOF's function to meet the differing needs of its users. Each IOF user has his own user profile.

Your **user profile** is a set of parameters which determines the way you see the system and how you use IOF. You can modify your profile to suit your needs without interfering with the profiles of other users.

See Section 2.4 of this manual for more information about **user profiles**

### 1.2.5 Menus

**Menus** provide lists of options, and prompt you to select one of the options presented. There is one type of menu available for formatted screen terminals and another for serial terminals. They are accessed differently.

There is further information on **menus** in Section 2.2 of this manual.

### 1.2.6 Help

**Help texts** are available throughout IOF; all you have to do is enter a question mark in the appropriate field or, when working in line mode, enter a question mark before the relevant command or parameter name. Commands, keywords, menus, prompts, and operating contexts have corresponding help texts. These texts can help you discover various functions, or remind you of forgotten features or information.

Help texts are discussed in Section 2.3 of this manual.

## Introduction

### 1.2.7 GCL Commands and Directives

**GCL** (GCOS 7 Command Language) is a set of commands through which you can communicate with the system. You can use these commands to request the system to carry out activities such as executing a program, copying a file, or canceling the execution of a batch job. You can enter commands interactively at the terminal, store them in a file for multiple executions, or compile them into procedures that create new GCL commands.

A **directive** is a special type of GCL command. In contrast to other commands, which you can enter only in specific contexts, you can issue a directive any time the system requests input.

The use of GCL procedures is discussed in Section 3.6 of this manual.

## Getting Started With IOF

### 1.3 INFORMATION YOU NEED TO KNOW TO USE IOF

Before you can use IOF, you must be registered as an authorized user and have an entry in the system's catalog with attributes. This allows you to access IOF. For this purpose, contact your System Administrator who will provide you with the following information:

- a user identification (usually your name)
- a password (optional)
- a project
- a billing

User name, Password, Project, and Billing are DSA (Distributed System Architecture) terms. Although they apply to IOF, they are independent of it. These 4 terms are discussed below.

#### 1.3.1 User name

**User name** identifies the terminal user within the network. It is generally the actual name of the user, although it can require truncation (it can only have 12 characters), or modification so that two users with the same name are not simultaneously connected in a network. It is the minimum information required to access the system, because all other information can be derived from the user name through default values that are preregistered in the system's catalog.

#### 1.3.2 Password

A **Password** is a personal and confidential code that can be used to prevent unauthorized access to the system. Passwords are optional, but their use is strongly recommended.

## Introduction

### 1.3.3 Project

A **Project** is a means of controlling access rights. Access to system objects such as commands, files, libraries, or working environments is granted to projects and not to individual users. When working under a project, you (the user) have all the rights and prerogatives associated with that project, but no other. Each user has a default project. When a user wishes to work under an authorized project which is not his default, this must be supplied in the log-on dialog.

<b>NOTE:</b> Depending on the restrictions setup by your system administrator, the commands available or accessible to you may be different from the examples and commands in this manual.
--

### 1.3.4 Billing

**Billing** is an accounting concept. It is the name of an account to which all resources consumed during a session are to be charged. Each project has a default billing, but it can have additional billings if it is large in scope. If the billing to be used in a session is not the default one for the project, it must be supplied as part of the log-on dialog.

## Getting Started With IOF

### 1.4 IOF IN A DPS 7000 INSTALLATION

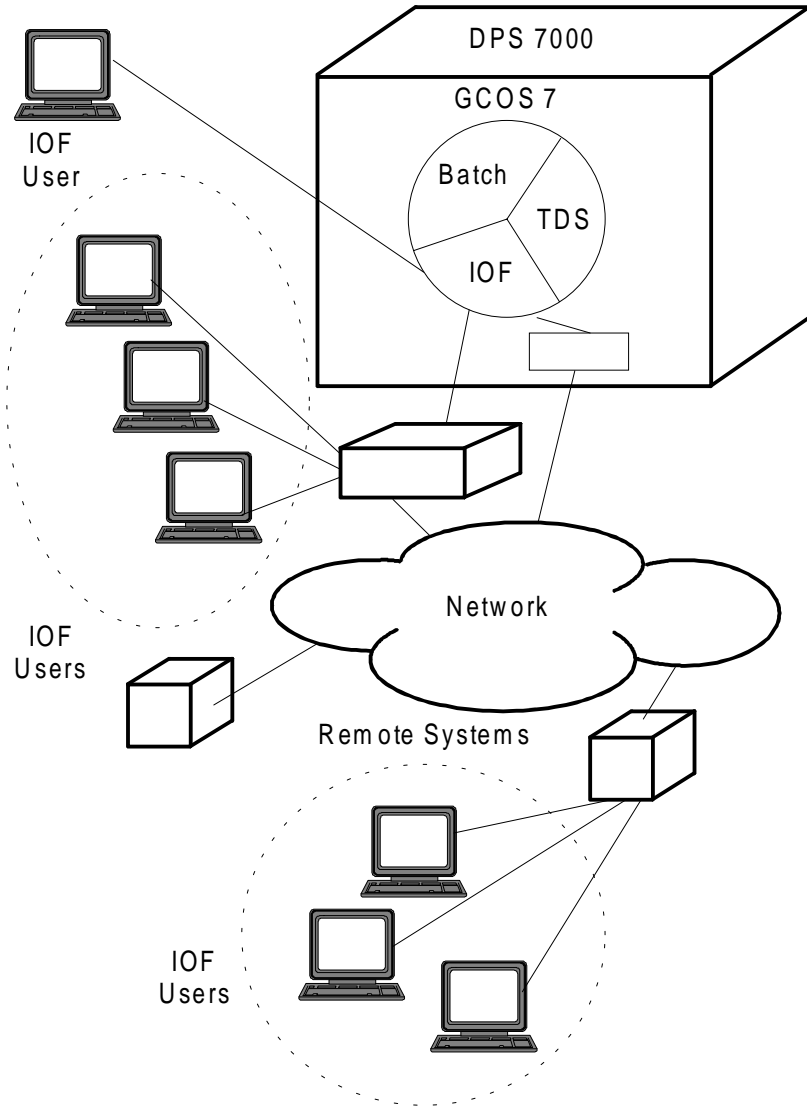
Figure 1-2 on the following page shows 3 of the ways a user can be connected to IOF:

- a single terminal connected directly to a DPS 7000
- a group of terminals connected through a front end processor
- a group of terminals connected through a network

Your IOF log on procedure depends on what type of connection you have.



Introduction



**Figure 1-2. IOF Connections**

Getting Started With IOF

## **2. Connecting to IOF**

There are a number of methods for connecting to IOF (for example through OPEN-TEAM, WIL7, etc.). These methods are generally automated to one degree or another. However, all the different connection possibilities are not discussed in this manual, contact your system administrator if you need to use one of the methods not covered.

The examples on the following pages address one method of connecting.

### **2.1 LOGGING ON**

As mentioned in Chapter 1, how you log on depends on how you are connected to IOF. A detailed description of one method of logging on to the network software and IOF (with default or specified options) is given below.

## Getting Started With IOF

### 2.1.1 Logging on to the Network Software

1. Power on the terminal and the modem.
2. Asynchronous Connection  
(Switched lines only) Dial up the Front-End Processor (FEP) and press the appropriate key to establish connection.

Press the H key to send the character by which the FEP recognizes the current speed of the line.

3. Synchronous Connection

(Switched lines only)

- a - dial up the FEP,
- b - press the appropriate key to establish connection,
- c - switch the modem to data mode.

Enter any character and press the transmit key.

4. Reply to the terminal type prompt:

MODEL:

with the appropriate terminal identifier (supplied by your System Administrator). If the line is dedicated to just one type of terminal, this prompt does not appear.

5. The FEP replies with the following messages:

```
*** DNS * RLS:XXXXXX* PAT:XXXXX/XXXXX/XXXXX * SYS:XX-XX-XX ***  
YOUR ID IS XXXX SC: XXXX MODEL:XXXXXXXX MB: XXXX
```

## Connecting to IOF

6. When the prompt `$$` appears, you log on using the connect command (CN).

CN is used to pass:

- the operational options to the application
- and the catalog parameters

**Note:** Catalog parameters are not discussed in this manual.

If you want to use application options other than those defined for the application at the site, enter these options immediately after you identify the application and the system, as follows:

```
$$CN <correspondent-name> [options]
```

Figure 2-1 is an example of what comes up on your screen when you log on.

```
4.  $$ 4200 MODEL: DKU107
5.  $$ 0000 *** D.N.S. * RLS:V4U1-7* PAT:OET20/1ET20/2ET20 * SYS: 1011B909
    $$ 1000 YOUR IDENTIFICATION IS 5987 SC: B909 MODEL:DKU7107 MB: 5987
6.  $$ 5800 YOU ARE REGISTERED
    $$cn -dmb iof -sc bf84 -usr jones          -pw mifo
```

**Figure 2-1. Example of a Log-on Screen**

## Getting Started With IOF

### 2.1.2 Logging on to IOF with Default Options

To log on to IOF, enter:

```
*$CN IOFcorrespondent-name [IOF-options]
```

The correspondent name specifies the name of the site to which you wish to connect. Ask your System Administrator for the appropriate correspondent name. IOF options are discussed below.

1. Either reply to the following prompts as you receive them:

```
$$ 1300 BILLING:  
$$ 1400 PERSON:  
$$ 1500 PROJECT:  
$$ 1600 PASSWORD:
```

or enter your user identification parameters directly in the \$\*\$CN command:

```
-BA <billing>  
-USR <user>  
-PJ <project>  
-PW <password>
```

To pass parameter values to the application, enter:

```
-STR <string>
```

You can enter **all** the operational options using the following format:

```
*$CN <application-name><system-name>  
-USR <user>  
-PJ <project> -PW <password>  
-BA <billing> <options>
```

## Connecting to IOF

2. After you log on to IOF, the system issues the following message:

```
$$ 0100 YOU ARE CONNECTED TO IOF
```

You are now connected to IOF. If your connection is rejected, a diagnostic stating why - wrong password, unauthorized project, no IOF on the site, etc. - is displayed. In case of rejection, you get the initial prompt again (that is, you are back to step 6, in Section 2.1.1).

## Getting Started With IOF

### 2.1.3 Logging on to IOF with Specified Options

#### IOF OPTIONS

Many of the concepts discussed in this manual are general concepts that apply to communication with all Bull systems or applications. In the previous discussion about logging on to the Network and IOF, information specific to GCOS 7 and IOF is identified as "IOF-options." These options are specified in an option string as the last part of the log-on sequence.

The general form of an IOF option string is as follows:

```
[$station-name] [![NS] [NM] [NEW] [*environment]]
```

#### **Example:**

```
$STN3  
!NS *ENV3  
!NS NM *ENV3  
$ST2 !*ENV3  
!NEW *N6
```

The option string has 5 possible components:

- Station
- No startup
- Mailboxes
- Restarting an Interrupted Session
- Environment

If you specify more than one component, you must protect the set of components with double quotes (" ").



## Connecting to IOF

### **Example:**

```
$*$CN IOFH222 -STR !*ENV3  
$*$CN IOFH222 -STR "$STN3 !NEW *ENV3"
```

or the equivalent,

```
$*$CN -MB IOF -SC H222 -STR !*ENV3  
$*$CN -MB IOF -SC H222 -STR "$STN3 !NEW *ENV3"
```

Note that if you specify an environment, it must always be the last element in the IOF option string.

Three of the five possible components of the option string, **No startup**, **Mailboxes**, and **Restarting an Interrupted Session**, are discussed here.

### **No startup**

You can bypass the activation of optional startup sequences. To do so, you must specify NS (No Startup) as part of your log-on messages. For example:

```
$*$CN IOFH222 -STR !NS  
$*$CN IOFH222 -STR "$ST3 !NS NEW"
```

or the equivalent,

```
$*$CN -MB IOF -SC H222 -STR !NS  
$*$CN -MB IOF -SC H222 -STR "$ST3 !NS NEW"
```

See Section 2.1.5 later in this Chapter for more details about startup sequences.

## Getting Started With IOF

### Mailboxes

You (every IOF user) are provided with a mailbox. This acts like a conventional mailbox; various messages can be stacked in your mailbox even if you are not connected. Messages that are stacked in your mailbox can come from two sources:

- other users, who send messages to you
- the system, to inform you of some event, such as the termination of a job that you have submitted

When you log on to IOF, the first thing the system does is list the contents of your mailbox, displaying all the messages that have been stacked since you last read your mail. You can bypass your mail. To do so, you must specify NM (No Mail) as part of your log-on message. For example:

```
$*$CN IOFH222 -STR "$ST3 !NM"  
$*$CN IOFH222 -STR "!NEW NM"
```

or the equivalent,

```
$*$CN -MB IOF -SC H222 -STR "$ST3 !NM"  
$*$CN -MB IOF -SC H222 -STR "!NEW NM"
```

### Restarting an Interrupted Session

If your IOF session is interrupted by a communication failure or some other event, you can reconnect to this interrupted session without any loss of information. To do this, you must reconnect within a certain time-period after the interruption. This time-period is defined at installation, and is the responsibility of the System Administrator or Operator.

## Connecting to IOF

If you **do not** want to reconnect to your interrupted session, use the NEW keyword as part of your log-on reply to indicate that you want to initiate a NEW session. For example:

```
*$CN IOFH222 -STR "$STN3 !NEW"  
*$CN IOFH222 -STR !NEW
```

or the equivalent,

```
*$CN -MB IOF -SC H222 -STR "$STN3 !NEW"  
*$CN -MB IOF -SC H222 -STR !NEW
```

If you do not use NEW, and your last session was interrupted, the system issues the following message:

```
YOUR LAST SESSION HAS BEEN RECONNECTED TO THIS  
TERMINAL
```

Any IOF options you have specified are ignored. The options that apply are those that were specified in the preceding connection.

## Getting Started With IOF

### 2.1.4 Resolving Difficulties

#### Error messages

Listed below are the principal error messages received during connection.

\$\$1821            Node not accessible.

1800            Disconnection requested by  
Line out, Datanet down...

1200            CONNECTION REFUSED:

(The following codes can appear in conjunction with \$\$1200.)

000B            Not enough system resources for IOF  
(tables).  
Maximum number of IOF jobs reached.  
Incorrect user information.

000C            User already connected to IOF and DJP.

000D            Q Class not started or not operational.  
ORACLE MI not available.

0018            Security check fails.

0042            User already connected.

Consult your system administrator about any other error messages.

## Connecting to IOF

### 2.1.5 Using Startups

In general, a startup sequence is a set of command entries that run automatically each time you log on. IOF's startup facility allows a stored sequence of GCL commands to be executed automatically at log on.

The execution of startup sequences can be mandatory or optional. You can bypass an optional startup sequence, but only the project SYSADMIN (System Administrator) can bypass mandatory sequences.

One of the major functions of a startup sequence is to set the appropriate user profile (see Section 2.4 in this Chapter for more information about user profiles). A typical use of a startup sequence is for you to be placed under the control of a particular processor without having to request it. You can store all the commands required to put you directly in your normal operating context in the startup sequence .

IOF startup sequences can be defined at three levels:

- system                                 SITE\_I
- project                                 projectname\_I
- user                                     projectname\_username\_I

Startups are found in the SITE.STARTUP library. Two example startup sequences are presented on the next page.

## Getting Started With IOF

The following is an example of a user level startup sequence for DEV\_JONES\_I.

```
SEND '***STARTUP JONES***';
SEND ' ';
SEND ' ';
LET # #CAT ('TODAY IS ',#DATE);
SEND ' ';
MDP WD=DEV.TEAM1;
MWINLIB BIN .BIN;
MWLIB BIN .BIN;
MWINLIB SL .SL;
MWLIB SL .SL;
LET # #CAT('Your input BIN library is', #BINLIB);
LET # #CAT('Your output BIN library is', #BLIB);
LET # #CAT('Your input SL library is', #SINLIB);
LET # #CAT('Your output SL library is', #SLIB);
```

The following is an annotated example of a project level startup:

```
LET # 'BEGIN STARTUP';           } issue a message.
MWLIB SL .SL;                    }
.                                 } define the working
.                                 } libraries and other
MWINLIB BIN .BIN;                } common rules.
                                  }
                                  } then ask to execute
                                  } a user-dependent
AI #CVNAME(#CAT(#USERID,-STP));  } sequence built up
                                  } from the user's name
                                  } followed by "-STP".
LET # 'END STARTUP';           } issue a message.
```

## Connecting to IOF

In the second example, the first part of the sequence determines the rules common to all project users. In the second part of the sequence, each user puts what they want in their user-dependent sequence. For example, JONES-STP (STP stands for Startup) contains the command to enter IQS, while SMITH-STP contains her additional library definitions and MAIL OFF command, etc.

What you put in your startup is a matter of project organization, personal taste, habits, level of expertise, etc.

## Getting Started With IOF

### 2.2 USING MENUS

Depending on your installation, IOF can present you with a variety of messages once you have logged on:

- a welcome message
- message of the day
- the contents of your "mailbox"

IOF can also execute a "startup sequence," as discussed in the previous section of this Chapter (2.1.5).

Once your messages are displayed and the startup sequence has run, the system displays three plus characters (+++) at the bottom of the screen, and waits for a reply. This allows you as much time as you require to read everything that is displayed on screen. The next screen will not appear until you press **ENTER**. It is generally a menu screen.

An example of a menu screen is shown in Figure 2-2. This is a standard screen, your administrator may have modified the commands for your system.

A title that states the name of the current operating context is displayed at the top of the screen. To the left of the title, a screen number is indicated in the form p/n: where p is the current screen and n is the total number of screens.



## Connecting to IOF

```
1/15                                IOF
1 BIND_CU                            BD  bind compile units
2 BUILD_FILE                          BF  allocate disk file & declare attributes
3 BUILD_LIBRARY                       BLIB build a library
4 BUILD_SYSTEM                        BSYS build a system file
5 CBL                                 COBOL compile COBOL-85 program(s)
6 CHECH_CATALOG                      CKCAT CHECK CATALOG STRUCTURES
7 CLANG                               CL   compile C program(s)
8 CLEAR_FILE                          CLRFB erase contents of a file
9 CLEAR_LIBRARY                       CLRLIB erase contents of a library
10 CLEAR_VOLUME                       CLRVB delete all files on a volume
11 COMPARE_FILE                       CMPFB compare contents of two files
12 COMPARE_FILESET                    CMPFST compare a fileset with a refernce file
13 CONNECT_APPLICATION                CN   connect IOF application
14 CONVERT_FORM                       CVFORM convert GPL forms into C language
15 COPY_CATALOG                      CPCAT copy a catalog file into another one
16 COPY_FILE                          CPF   copy file into another of the same type
17 COPY_FILESET                      CPFST copy filset into another of the same type
18 CREATE_CATALOG                    CRCAT create a catalog

-->:___
-: _____
-: _____
+++
```

**Figure 2-2. Example of a Menu Screen**

The title and screen number are followed by up to 18 lines (seen at the top of the screen in Figure 2-2) which constitute the actual contents of the menu.

Each line consists of a number (from 1 to 18), and a name, sometimes followed by an alias name (an abbreviation) and a short explanatory text.

At the bottom of the screen there is a blank selection field of three characters, introduced with a right-pointing arrow. There are also two blank lines headed with the -: characters. Finally, there is either a blank line or, if the current screen is not the last screen of the menu, a line with three plus characters (+++).

## Getting Started With IOF

The menu is provided to prompt you to select ONE of the options presented.

To make a selection you sometimes need to page through the different screens of the menu or ask for assistance. This is the purpose of the selection field marked with the -->: arrow. In that field you can enter:

- a number from 1 to 18, to indicate your selected option
- a question mark (?) to ask for help, see Section 2.3 of this Chapter
- a greater than (>) character (or leave it blank) to go on to the next page
- a less than (<) character to go back to the preceding page
- >n or <n to move respectively n pages forwards or backwards
- =n to move to page number n

Any erroneous input is highlighted on the screen. You can then correct the input as required and submit it again.

To move from field to field in the IOF menus and screens, use the **TAB** key to go forward, and **SHIFT/TAB** key to go backwards.

## Connecting to IOF

### 2.2.1 Executing Commands

When you are just getting started with IOF, executing commands through menus is the best method for you. As mentioned in Chapter 1, you can use commands to request that the system execute a particular program, copy a file, cancel a batch job, etc.

Figure 2-3 is an example of executing the FSE command. The FSE command calls the Full Screen Editor. There is a whole section devoted to using FSE in Chapter 3 (Section 3.2).

In this example, the selection field (-->) is used to select the name of the command by its number (17), as indicated on the screen. Note that the FSE command is found on the 3rd screen of the main IOF menu. This is indicated by the number 3/15 you can see in the upper left corner of Figure 2-3.

The two lines at the bottom of the screen are used to enter parameters. See the next section of this Chapter, 2.2.2, for more information about parameters.

To execute your command, press **ENTER**.

If your entry is erroneous, you are prompted with a screen to help you correct it.

**Note:** If you are an experienced user, and are working in line mode, you can enter multiple commands. You must separate them by a semi-colon (;). You can use a trailing semi-colon (for example, 9;7;), but it is not mandatory.

## Getting Started With IOF

```
3/15          IOF
1 DELETE QUOTA FILE      DLOF  delete SITE.QUOTA file
2 DUMPJRNL              run the DUMPJRNL utility
3 EDIT                  ED     call the line editor
4 ENTER_DIUF            DIUF   DNS-CNS installation & updating facility
5 ENTER_GIUF            GIUF   GCOS7 INSTALLATION AND UPDATING FACILITY
6 ENTER_GSF             GSF    GSF Installation Updating and Usage
7 ENTER_RMOS            RMOS   Remote Multiplexed Operator Support
8 EXEC_PG               execute a user program
9 EXECUTE_DELTE_REMOTE_FILE delete a remote file
10 EXECUTE_FILE_TRANSFER EXFT   perform a synchronous file transfer
11 EXECUTE_MODIFY_REMOTE_FILE modify a remote file
12 EXECUTE_SCRIPT        ESCR   execute script request
13 EXPAND_FILESET        EXPFST expand a fileset expression
14 FILL_MIRROR           FLMIR  duplicate contents of mirror alone disk
15 FOR77                 F7C    compile FORTRAN-77 program(s)
16 FPG7                  MNFPCONF call the fep configurator utility
17 FSE                   call the Full Screen Editor
18 GENERAL_TW_FACILTIY   GTF    GTWriter transaction facility
-->:_17
-:_____
-:_____
+++
```

**Figure 2-3. Executing a Command from a Menu**

### 2.2.2 Supplying Parameter Values

Sometimes extra information, in the form of parameter values, is required to execute a command. One or more parameters can need defining. There are 3 types of parameters:

- mandatory
- optional with default
- optional without default

You must define a **mandatory parameter** in order to execute the command. You do not have to define an **optional parameter** that has a **default value**, the parameter is defined for you by a predesignated value. Neither do you have to define an **optional parameter without default** value, the parameter is left undefined if you do not supply a value.

Sometimes there is more than one screen of parameters for a given command. When this is the case, it is indicated in the same manner as for menus and help texts (see Section 2.3.1 of this chapter).

In the example given below, you can supply extra information. The screen on the next page appears when you call FSE in menu mode. One parameter field is shown:

LIB (output library)

You can leave the parameter blank because there is a default value. However in this example we supply the value, as shown in Figure 2-4.

## Getting Started With IOF

```
1/1          FSE          -->:___  
  
          call the Full Screen Editor  
  
LIB          output library (default is #SLIB)  
DEV.TEAM1.SL  
  
-----
```

**Figure 2-4. Example of Supplying Parameter Values**

Enter the parameter value in the field provided. In this hypothetical example, you enter DEV.TEAM1.SL for (LIB) the output library.

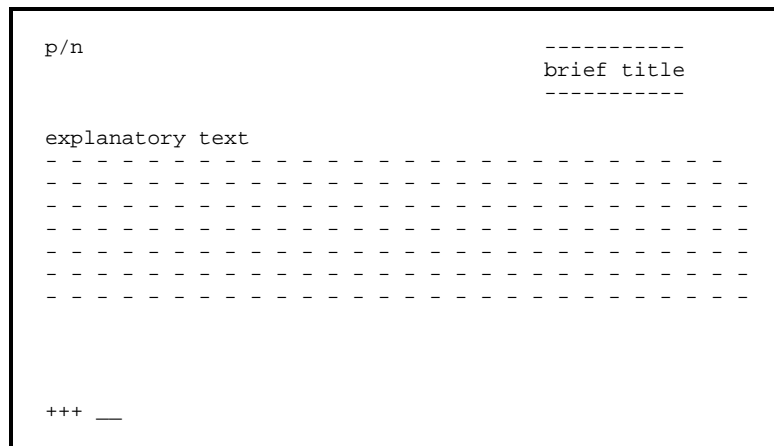
## 2.3 GETTING HELP

### 2.3.1 Help Texts

To obtain a Help text, just enter a question mark (?) in any one of the input fields in screens other than Help texts, and then press **ENTER**.

To obtain help for a command, type in a question mark immediately followed by the command name (?command)

The general layout of a Help screen is as follows:



**Figure 2-5. Help Screen Layout**

## Getting Started With IOF

The top of the screen shows a screen number in the form p/n (screen p out of a total of n) and a brief title for the Help text. An explanatory text on the appropriate subject follows.

When you see three plus (+++) characters at the bottom of the screen, this is an indication that there is more Help text (at least another screen). When the (+++) characters do not appear, there are no further screens.

There are a number of operations available with Help text screens. To use them, enter the appropriate character in the entry field that follows the (+++) characters at the bottom of the Help screen.

Some potentially useful operations and their corresponding characters are as follows:

- enter a greater than (>) character or leave blank to move to the next page
- enter a less than (<) to move to the preceding page
- enter >n, =n to move respectively n pages backwards, n pages forwards or to page number n
- enter a slash (/) character to return to the place where the Help text was requested
- enter a question mark (?) to get another Help text which explains what is being explained here

Any other entry is reported as an error.

**Note:** Help texts are not a substitute for paper documentation. Help texts provide immediate, on-the-spot assistance for those people who have forgotten some point, or need some further information. Help texts do not provide a basic general knowledge of IOF.



## Connecting to IOF

### 2.3.2 An Example of Getting Help

To get help with the LIB parameter, enter a question mark (?) in the space provided for the parameter value. Compare Figure 2-6 below, to Figure 2-4 (Section 2.2.2 Supplying parameter values).

```
1/1                FSE                --->:___
                call the Full Screen Editor

LIB                output library (default is #SLIB)
?
```

---

**Figure 2-6. Requesting Help for the LIB parameter**

## Getting Started With IOF

Press **ENTER**.

The following Help text giving you information about the LIB parameter is displayed:

```
1/1                                -----  
                                LIB  
                                -----  
  
The LIB parameter specifies the SL Library to be used as  
the output library.  
  
You can supply any valid library description  
  
If you omit the LIB library whose description is contained  
in the system variable #SLIB is assigned (#SLIB can be set  
and modified by the MWLIB command.) If #SLIB has not been  
set, a temporary SL library (TEP.SLLIB$TEMPRY) is used.  
  
Examples:  
  
LIB=PRJBR.MYLIB  Use the cataloged library PRJBR.MYLIB.  
  
LIB=DOCU$RES     Use the uncataloged resident library DOCU.  
  
LIB=WS:D1706:MS/M452  
  
                Use the uncataloged library WS which  
                resides on the MSU0452 volume named D1706.
```

**Figure 2-7. LIB parameter HELP text**

## Connecting to IOF

### 2.4 USER PROFILES

Different users have different requirements: some are novices, while others are experienced; some appreciate being assisted by prompts, while others find it annoying; some prefer to enter lines in free format, while others prefer line format, etc. These are the types of preferences and information that go into a user profile.

You (and every user) have your own profile. You can modify your profile without affecting the profiles of other users. Your profile indicates to the system which options are to be supplied for you. The most important elements of your profile are listed below:

- the working directory, the directory in which most of your activity is centered
- the mode, you have a choice of 3 modes to work in:
  1. novice mode: in this mode you are systematically presented with menus for entering commands
  2. menu mode: in this mode, you only get a menu when you specifically request one, or when you make an error (this mode is not recommended for novice users)
  3. expert mode: in this mode, you never get any menus or prompts, only error messages if an error occurs.
- the user's national language, which determines the language in which the Help texts are displayed
- other items not covered in this manual

## Getting Started With IOF

Use the MODIFY\_PROFILE (MDP) directive to modify your user profile. Following are some examples of its use:

```
MDP WD=DEV.TEAM2      {set working directory to DEV.TEAM2}
MDP NOVICE=0 MENU=1  {change to menu mode}
```

The DISPLAY\_PROFILE (DP) directive displays the elements of the user profile together with additional information on the operating environment (project, billing, terminal type, etc.).

**Note:** Remember, a **directive** is a command that can be issued at any time when the system requests input.

The following (Figure 2-8) is an example of what is shown on screen in response to a DISPLAY\_PROFILE directive given in line mode.

```
S: DP
11.38      DATE = THU DEC 22, 1994  WEEK=451
USERID=JONES BILLING= PROJECT=DEV  RON=2441
MODE=IOF  CPU=0.074 ELAPSED=32.003
WD=DEV.TEAM2 WENVT=          WSTATION=BN75  LSYS=BN75
NOVICE=0 MENU=1 GCLFORM=LINE GCLMODE=GCL LANG=0 FORMLANG=COBOL
TTYPE=DKU7105 PW=80 PL=24 PAGERMODE=1 ROLL=0 AUTOLF=1 CSET=PLW
D1=$$ 00= INVCHAR=. DLCHAR= EDITCTL=( $ [ ] OK=1
YES=(YES Y 1) NO=(NO N 0) BRKPMODE=0 CONFIRM=0
TABS=
EXPTABS=1 DEBUG=0 TRACE=v
NO DEFAULT LIBRARY
11.38      LOG IS NOT ACTIVE
S:
```

**Figure 2-8. Example Response to a DISPLAY\_PROFILE Directive**

## Connecting to IOF

### 2.5 LOGGING OFF

To end an IOF session, select the option BYE, or, in serial prompting mode, enter:

BYE ;

the system then prints some statistics in the following form:

```
<<<09.17.33 JONES LOGGED OFF
<<< CPU 0.161
<<< ELAPSED 12.980
<<<
```

where CPU is the total time, in minutes, that the central processor unit was used, and ELAPSED is the duration of the session being closed.

## Getting Started With IOF

### 2.6 OUTLINE OF AN IOF SESSION

An IOF session essentially involves logging on, performing some operations, and then logging off.

The following is an outline of a very simple, hypothetical interactive session in novice mode, where you:

- log on
- execute a command(s)
- log off

#### 2.6.1 Logging On

Power on your terminal and log on to IOF as explained in Section 2.1 of this Chapter. The system then issues the following message:

```
$$ 0100 YOU ARE CONNECTED TO IOF
```

IOF identifies itself; issuing a welcoming message

```
>>>08.35 IOF-D456 etc
```

The "Message of Today" from the system operator appears next.

```
MOT: THE SERVER WILL BE SHUT DOWN ON FRIDAY  
APRIL 04.
```

Then, any mail that you might have waiting is displayed.

```
-->ISSUED ON MAR 30, 1994  
-->19.32 X2865 OUTPUT COMPLETED ABSENTEE BETH  
-->ISSUED ON APR 2, 1994  
-->09.05 FROM SMITH: Don't forget our  
conference call Monday AM.
```

Once you have read all your messages press **ENTER** to display the next screen.

## Connecting to IOF

**Note:** If you have **more** messages they are displayed at this time.

If you have **no more** messages, a menu is displayed.

You have no more messages. The main IOF menu, similar to the following one in Figure 2-9, is displayed.

```
1/15          IOF
1 BIND_CU      BD   bind compile units
2 BUILD_FILE   BF   allocate disk file & declare attributes
3 BUILD_LIBRARY BLIB build a library
4 BUILD_SYSTEM BSYS build a system file
5 CBL          COBOL compile COBOL-85 program(s)
6 CHECH_CATALOG CKCAT CHECK CATALOG STRUCTURES
7 CLANG        CL   compile C program(s)
8 CLEAR_FILE   CLRF erase contents of a file
9 CLEAR_LIBRARY CLRLIB erase contents of a library
10 CLEAR_VOLUME CLRV delete all files on a volume
11 COMPARE_FILE CMPF compare contents of two files
12 COMPARE_FILESET CMPFST compare a fileset with a reference file
13 CONNECT_APPLICATION CN connect IOF application
14 CONVERT_FORM CVFORM convert GPL forms into C language
15 COPY_CATALOG CPCAT copy a catalog file into another one
16 COPY_FILE   CPF   copy file into another of the same type
17 COPY_FILESET CPFST copy filset into another of the same type
18 CREATE_CATALOG CRCAT create a catalog

-->:____
-:_____
-:_____
+++
```

**Figure 2-9. Example of a Menu**

## Getting Started With IOF

### 2.6.2 Executing a Command

Choose a command from the main menu (Figure 2-9). Type its corresponding number in the selection field.

Press **ENTER**

If any parameter screens are displayed, define the necessary values. Press **ENTER** to execute the command.

If no parameter values need to be defined the command is executed the first time you press **ENTER**.

You can consecutively execute more commands if you wish.

### 2.6.3 Logging Off

Now that you have executed your operation(s), you can log off.

To end your IOF session, select the option BYE.

The system then prints some statistics. (For more details, see Section 2.5 of this Chapter.)



## 3. Developing Applications

### 3.1 SELECTING DEFAULT LIBRARIES

Remember, a library is a file organized into a series of subfiles called members. Each member contains a separate unit of information. The use of library members makes it easy to retrieve, edit, and store the various types of information that you need to use in your daily work.

You must declare your default or working libraries for every session.

You do this using the MWINLIB and MWLIB commands discussed below. You must have a working directory to do this in, and generally the System Administrator sets this up for you.

There are 2 basic types of libraries:

- input libraries
- output libraries

Both input and output libraries can, in turn, be any one of 5 different types of library:

- Source Language (SL)
- Compile Unit (CU)
- Load Module (LM)
- Sharable Module (SM)
- Binary Object (BIN)

**NOTE:** For more information about these types of libraries, refer to Chapter 4.

## Getting Started With IOF

In this chapter, we use Source Language, or SL, libraries in the examples.

At any given moment during your current session you can define up to three input SL libraries, and one output SL library. However, you are not required to have more than one of each.

As a group, your input and output libraries are called your working libraries.

You can include the names of your working libraries in your startup sequence (see Section 2.1.5 of this manual). This means that when you log on your working libraries are ready for use.

### 3.1.1 Input Libraries

An input library is a reference library. You can read the members of an input library, but you cannot write to them.

You define or redefine an input library with the command MWINLIB. You enter the MWINLIB command with the following syntax:

```
S: MWINLIB [<libtype>]INLIB=<library-names>;
```

INLIB            is the name of the library you want to define.

You can assign all 3 of your input libraries at the same time. You must put the library names in parentheses and separate each one by a comma and/or one or more spaces. The following is an example:

```
S: MWINLIB SL INLIB=(DEV.TEAM1.SL , DEV.TEAM2.SL);
```

INLIB            is the name(s) of the Source Language (SL) library(ies) you want to define.

## Developing Applications

### 3.1.2 Output Libraries

The library that you work in and create members for is your output library. You can both read and write to output libraries.

You define or redefine your output library with the MWLIB command. You enter the MWLIB command with the following syntax:

```
S: MWLIB [<libtype>] LIB=<library-name>
```

LIB is the name of the library you want to define.

**Note:** It is often easiest to have the same library for both your input and output libraries. To do this you simply enter the same library name when you define your input and output libraries.

The following is the an example of declaring an output library:

```
MWLIB SL LIB=DEV.TEAM2.SL;
```

## Getting Started With IOF

### 3.1.3 Output Library Listing

It is possible to print out a report on what you have done in the input/output libraries.

You define or redefine your default listing output library with the MPRTLIB command. You enter the MPRTLIB command with the following syntax:

```
S: MPRTLIB PRTLIB=<library-name>
```

PRTLIB is the identification of the listing output library. When you omit PRTLIB, the listing output library is canceled (that is, becomes unspecified).

The following is an example of defining a default listing output library:

```
MPRTLIB PRTLIB=DEV.TEAM2.SL;
```

## Developing Applications

### 3.2 USING FSE

As mentioned earlier, in Section 1.2.3 of this manual, there are two products available for program entry and editing: EDIT and FSE (Full Screen Editor). EDIT is oriented towards serial terminals and batch usage. FSE is discussed in this manual because it is more user-friendly.

You can activate FSE as a stand-alone processor, or from within MAINTAIN\_LIBRARY (and certain other processors) as a command.

You use FSE to edit and enter texts from full-screen terminals. You can refer to lines in the same manner as in EDIT. However, full screen facilities are provided whereby you enter modifications directly on the screen using the cursor and local editing keys. This makes editing much easier because you can see each modification in the context of its surrounding lines.

This section provides only a brief introduction to some basic FSE functions. Further information about using FSE is available in the following Bull publications:

*Full Screen Editor User's Guide*  
*Library Maintenance Reference Manual*  
*Library Maintenance User's Guide*

## Getting Started With IOF

### 3.2.1 Calling FSE

You can call the Full Screen Editor in either line mode or novice mode.

The Full Screen Editor command is FSE.

#### **Novice Mode**

To call the Full Screen Editor in novice mode, you must type the command FSE or its corresponding number in the appropriate field. (FSE is listed in the menu like the other IOF commands, and you execute it in the same manner.)

Press **ENTER**.

If you are in novice mode, the system displays the screen you can see in Figure 3-1 on the next page.

## Developing Applications

```
1/1          FSE          -->: __
          call the Full Screen Editor

LIB          output library (default is #SLIB)
DEV.TEAM1.SL
```

---

**Figure 3-1. Library Parameter Screen**

In this screen you can indicate the name of the output library in which you want work.

However, the output library has a default value here #SLIB, and thus is optional. In this example we use the output library named DEV.TEAM1.SL.

Once you type in the LIB value and press **ENTER**.

The system then displays the initial FSE screen, Figure 3-2.

The initial FSE screen is a menu. In this example there are 8 screens of commands, but the number can vary. The first screen lists some of the most commonly used FSE commands.

## Getting Started With IOF

**Note:** Like everywhere in IOF, to get a Help screen simply enter a question mark (?) in any field. To get help with a command, enter the name of the command followed immediately by a (?) in the first selection field line that runs across the bottom of the FSE menu screen. For example: CREATE?.

```
1/8          F_S_E
1 COPY      CP      copy member(s) from input to output lib
2 CREATE    CR      create a member in output lib
3 DELETE    DL      delete member(s) in LIB
4 DISPLAY   D       display assigned libraries
5 INLIB1    IL1     assign or modify INLIB1
6 INLIB2    IL2     assign or modify INLIB2
7 INLIB3    IL3     assign or modify INLIB3
8 LIB       SLLIB   assign or modify LIB
9 LIST      LS      list directory of input or output lib
10 MODIFY   MD     modify member with result in output lib
11 PRINT    PR     print a member of input or output lib
12 QUIT     Q      leave FSE
13 RENAME   RN     rename a member in lib
14 RENUMBER RB     renumber member(s) of LIB

-->:___
:_____
:_____
+++
```

**Figure 3-2 initial FSE Menu**

Most FSE screens have either a selection field in the upper right corner, denoted by -->: or, an EXIT field at the top. To quit an FSE screen, you enter a slash (/) character in either of these fields and press **ENTER**.



## Developing Applications

There are 2 ways to choose an option (remember, you can only choose one at a time).

- You can enter its number in the selection field in the lower right corner of the screen, designated by the -->: sign, and press **ENTER**.
- Otherwise, you can enter its name or abbreviation in the first of the two selection field lines that run across the bottom of the screen, and then press **ENTER**.

### Line Mode

If you want to access FSE from line mode, you enter the command followed immediately by the name of your output library at the S: prompt, as follows:

```
S: FSE <output-library-name>;
```

The equivalent for the example just given in novice mode is

```
S: FSE DEV.TEAM1.SL
```

The system then displays the prompt:

```
F:
```

This is the point at which you enter your FSE commands. We continue in novice mode in this section, but all the commands explained in novice can be used in line mode as well as.

## Getting Started With IOF

### 3.2.2 Creating a Subfile or Member

You use the CREATE command to create a subfile, or member, as it is called here. Type in the number, name, or abbreviation of the command in the appropriate field. Then press **ENTER**.

The following screen, Figure 3-3, is displayed.

1/1	CREATE	(+) -->:___
	create a member in output lib	
MEMBER	+ member to be processed <u>JONES-STP</u>	
LANG	language type	<u>DAT</u>
REPLACE	do you want to overwrite ?	<u>0</u>
APPEND	append in member	<u>0</u>
-----		

**Figure 3-3. FSE Create Screen**

In the member selection field, type a name for the member you want to create. In the example above, the member is called JONES-STP.

## Developing Applications

The plus (+) character next to the member field indicates that this a mandatory parameter. The member name can have up to 31 characters. You can use the alphabetic A-Z and the numerics 0-9. There cannot be any blank spaces in the member name.

There are several parameters with their corresponding selection fields, but they are not mandatory when you first create a member. When you have typed in the necessary information (in this case simply the member name), press **ENTER**. The following screen, Figure 3-4, is displayed.

```
H_FSE_CDAT  REC TXT EXIT__ NEXT__  JONES-STP  CREATE
**
**
01  _____
71  _____
141 _____
211 _____
-----
```

**Figure 3-4. Example of FSE Member Screen**

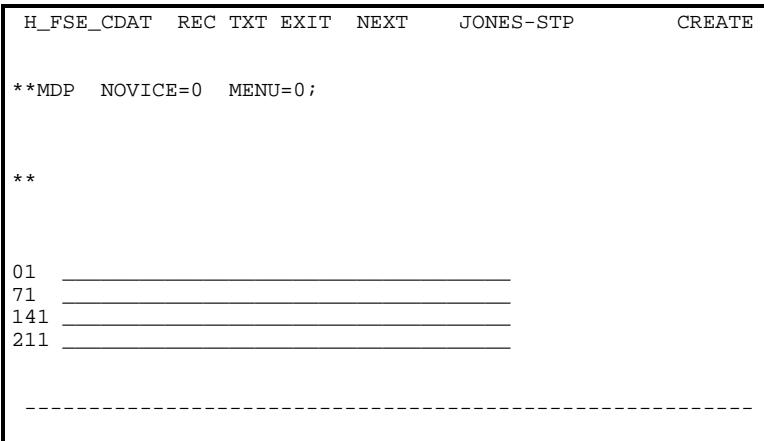
## Getting Started With IOF

The member name is displayed at the top of the screen, right of center. Further down the screen, on the left side, you can see a list of numbers, 01, 71, etc. This is where you enter your text. You start at 01. Each numbered field is 70 characters long. Press **ENTER** to end a line. If you do not press **ENTER**, a line can continue up to a total of 256 characters.

For example, to modify your user profile within FSE, you must use the MODIFY\_PROFILE (MDP) command. If you type the following:

```
MDP  NOVICE=0  MENU=0;
```

and then press **ENTER**, the following screen, Figure 3-5, is displayed.



```
H_FSE_CDAT  REC TXT EXIT  NEXT      JONES-STP      CREATE
**MDP  NOVICE=0  MENU=0;
**
01 _____
71 _____
141 _____
211 _____
-----
```

**Figure 3-5. Example of Entering Text in FSE Member**

Your text is entered, and has moved up the screen. You can now enter another line. You can repeat this as many times as necessary.

## Developing Applications

In this screen, Figure 3-6, you have created a second line.

```
H_FSE_CDAT  REC TXT EXIT  NEXT          JONES-STP  CREATE

** MDP NOVICE=0 MENU=0;

** MPRTLIB .SL;

01 _____
71 _____
141 _____
211 _____
```

**Figure 3-6. Example of Entering Text in FSE**

You have asked to define or redefine the default listing output library through the MPRTLIB command. (See Section 3.1.3 for more information about the MPRT command).

After you have created all the text you require, you want to leave CREATE to execute another operation. Type a slash (/) character in the EXIT field, and press **ENTER**, to quit the CREATE screen. Your member is saved automatically.

The system returns you to the initial FSE menu screen.

## Getting Started With IOF

### 3.2.3 Modifying a Subfile or Member

To modify your subfile or member, use the MODIFY command in the initial FSE menu. The procedure is the same as that for executing any command in FSE.

**NOTE:** Be sure you specify the name of the member you want to modify. If you simply enter the command without the name, your file is not loaded.

For example, type one of the following

```
MODIFY JONES-STP  
MD JONES-STP
```

in the command entry line at the bottom of the menu. Then press **ENTER**.

Your subfile is loaded directly, and the screen on the next page (Figure 3-7) displays your text.

## Developing Applications

```
b=MAIN m=LIB:JONES-STP DAT >ln=3          ctl____
                                         TOP
>>>> 10 MDP NOVICE=0 MENU=0;
      20 MPRTLIB .SL;
                                         BOTTOM

request          addr          param3
param1          param2
```

**Figure 3-7. Modify Member FSE Screen**

This screen (Figure 3-7) has three distinct areas:

- |              |  |
|--------------|--|
| Header area  | the first line of the screen which contains information about your member (subfile) and the control field.   |
| Text area    | the area where the text is found, and which is distinguished by TOP and BOTTOM, although whether BOTTOM is visible depends on how much text you have.  |
| Request area | the two lines at the bottom of the screen. This area allows you to use the text editing requests, which you enter in the request field, each of which has an address parameter (addr), and the fields to enter up to 3 additional parameters (param1, param2, and param3). |

Enter a ? in the request field to obtain a list of possible requests along with a brief description of each one.

## Getting Started With IOF

Suppose you want to change the FSE user profile from novice mode to menu mode, this involves changing line ten.

You type S, the SUBSTITUTE request, in the request field and an asterisk (\*), for all, in the address field. In the first parameter field you type the value you want to replace (MENU=0), and the second parameter field you type the substitution (MENU=1).

The screen is as follows (Figure 3-8):

```
b=MAIN m=LIB:JONES-STP DAT >ln=2          ctl ____
                                     TOP
    10 MDP NOVICE=0 MENU=0;
    20 MPRTLIB .SL;
                                     BOTTOM

request  S          addr  *
param1   MENU=0    param2 MENU=1    param3
```

**Figure 3-8. Example of Text Substitution in FSE**

Press **ENTER**.



## Developing Applications

You can see the results in the screen below, Figure 3-9.

```
b=MAIN m=LIB:JONES-STP DAT >ln=2          ctl ____
                                     TOP
    10 MDP NOVICE=0 MENU=1;
    20 MPRTLIB .SL;
                                     BOTTOM

request          addr
param1           param2           param3
```

**Figure 3-9. FSE Text Substitution Results**

## Getting Started With IOF

Suppose you now want to add lines to your text. Type A ( the APPEND command) in the request field. You want to add text on to the end of your existing text. You type 20 in the address field, so that the new text starts after line 20.

You can see the parameter values in the appropriate fields in the screen below, Figure 3-10.

```
b=MAIN m=LIB:JONES-STP DAT > ln=2          ctl ____
                                     TOP
10 MDP NOVICE=0 MENU=1;
20 MPRTLIB .SL;
                                     BOTTOM

request  A          addr  20
param1   param2    param3
```

**Figure 3-10. Requesting to Append Text**

Press **ENTER**.

There are other characters that have special functions. For example, the \$ character in the address field indicates the last line.

## Developing Applications

FSE displays the screen in Figure 3-11.

The cursor is positioned on the line just below line 20. You can add up to 4 lines.

```
b=MAIN m=LIB:JONES-STP DAT > ln=3          ctl _ ;
                                     TOP
10 MDP NOVICE=0 MENU=1;
20 MPRTLIB .SL;
    _____
                                     BOTTOM

request          addr
param1           param2          param3
```

**Figure 3-11. Appending Text in FSE**

Suppose you enter a third (and last) line of text in the text entry field shown in the screen above. Enter a semi-colon in the control field to indicate that you do not want to enter any further data. You then press **ENTER**.

## Getting Started With IOF

The results are seen in the following screen, Figure 3-12.

```
b=MAIN m=LIB:JONES-STP  DAT  > ln=3          ctl  ___
                                     TOP
      10 MDP NOVICE=0 MENU=1;
      20 MPRTLIB .SL;
>>>>20.1 LET # #CAT ('DATE: ',#DATE);
                                     BOTTOM

request  ___          addr
param1   ___          param          param3
```

**Figure 3-12. Results of Added Text in FSE**

You now want to renumber your text. To do so you must enter the RENUMBER command and its parameters.

## Developing Applications

Type RB (RENUMBER) in the request field. You want to renumber your entire text, so you type an asterisk (\*) in the address field. You want to renumber the text starting with line 10, in increments of 10. Type 10 in the param1 field (where to start renumbering) and 10 in the param2 field (by what increment to renumber).

The following screen, Figure 3-13, shows the parameter values in the appropriate fields.

```
b=MAIN m=LIB:JONES-STP DAT > ln=5          ctl ___1
                                     TOP
      10 MDP NOVICE=0 MENU=1;
      20 MPRTLIB .SL;
>>>>20.1 LET # #CAT ('DATE: ',#DATE);
                                     BOTTOM

request  rb          addr  *
param1   10          param2 10          param3
```

**Figure 3-13. Renumbering FSE Text**

Press **ENTER**.

## Getting Started With IOF

The screen below, Figure 3-14, shows the results of the renumbered text.

```
b=MAIN m=LIB:JONES-STP  DAT  > ln= 3          ctl  ___
                                     TOP
10 MDP NOVICE=0 MENU=1;
20 MPRTLIB .SL;
30 LET # #CAT ('DATE: ',#DATE);
                                     BOTTOM

request          addr
param1          param2          param3
```

**Figure 3-14. FSE Renumbered Text Results**

You have made all the required changes to your member (subfile), and have finished with FSE for the moment. Type SVQ in the request field and press **ENTER** to SAVE and QUIT.

This brings you back to the initial FSE menu. To quit FSE, type a slash (/) character in the selection field line at the bottom of the screen, and press **ENTER**.

### 3.3 CREATING AN SL LIBRARY

An SL library is a library that holds any kind of textual information. Examples of SL library members are letters, documents, and source programs.

**NOTE:** Now that you are more familiar with the system, the examples from here on are given in line mode.

You use the BUILD\_LIBRARY (BLIB) command to create an SL library. To create a library you must:

- give it a name
- declare values for several parameters

The general form of the BLIB command for an SL library is as follows:

```
S: BLIB LIB=<library-description>:media:dvc
      UNIT=<value>
      SIZE=<value>
      MEMBERS=<number-of-members>
      TYPE=<value>
      INCRSIZE=<value>
      MAXSIZE=<value>
      COMPACT=1;
```

The parameters are explained below:

**LIB** is the name of the library to be allocated, it states the volume and device class, or \$RES for resident libraries.

Please note that if the VOLSET facility is available, location syntax (media and device class, or \$RES) is optional. Refer to the Bull manual *Administering the Storage Manager*, for more information about the VOLSET facility.

## Getting Started With IOF

UNIT	is the unit of disk space. There are different values possible for UNIT, for example, CYL (cylinder), 100KB (100 kilobytes) and BLOCK, but they are not discussed in this manual.
SIZE	is the total amount of space (in the unit specified by the UNIT parameter) to allocate to the library. See Section 3.3.1 on allocating space.
MEMBERS	is the estimated number of members for this library.
TYPE	the type of library. The library can be one of the following types: Source Language (SL), Compile Unit (CU), Load Module (LM), Binary (BIN), Sharable Module (SM).  The default is SL.
INCRSIZE	is the increment by which the library size is increased when full, up to the maximum size specified by MAXSIZE. See Section 3.3.1 on allocating space.
MAXSIZE	Is the maximum amount of space allocated to the library. See Section 3.3.1 on allocating space.
COMPACT=1	specifies that trailing spaces in a line are compacted. This saves space overall for the library.



## Developing Applications

The following are 2 examples of using the commands to create SL libraries:

1. S: BLIB DEV.TEAM2.SL:MSDI01:MS/FSA SIZE=10  
MEMBERS=200

Where you create a cataloged SL library of 1 megabyte (10x100KB), with a directory for 200 members, on the FSA volume MSDI01.

Note that TYPE is not specified, and is thus the default (SL).

2. S: BLIB TEMPSL\$RES TEMPRY TYPE=SL SIZE=1  
MEMBERS=10

Where you create a temporary SL library on a resident volume.

### 3.3.1 Allocating Library Space

To allocate library space you must calculate the values of three parameters:

- SIZE
- INCRSIZE
- MAXSIZE

The length of a line of program code averages 100 bytes. You can use this figure to help you calculate the SIZE, INCRSIZE, and MAXSIZE values. For the purposes of the examples in this section (3.3), 1 unit equals 100kb.

#### SIZE

For example, a member that contains a 500 line program takes up 50,000 bytes. You estimate that this library will contain 20 members of approximately the same length (500 lines). Therefore, the total space you need to allocate for this library is approximately 1,000,000 bytes, or about 10 units. Thus, SIZE = 10.

#### INCRSIZE

For example, you estimate that the library may increase by as many as 10 members, each of the same length (50,000 bytes). Therefore the increment size is  $10 \times 50,000 / 5 = 100,000$  (here we predict 5 extensions of 2 members each). Thus, INCRSIZE = 1 (1 unit or 100Kbytes).

#### MAXSIZE

You calculate MAXSIZE according to the results of the above calculations. The maximum size of the library is reached after 5 extensions of the increment size. The calculation is  $10 + (5 \times 1) = 15$ . Therefore, MAXSIZE=15.

### 3.3.2 Creating a Directory

Remember that a directory is a group of files in the hierarchical tree structure. You can consider the directory itself a file, and that this file contains a list of object names or files.

IOF has a directory facility so that you have complete independence when you name files and objects. Within a given directory each file or object must have a different name, but objects in different directories can have the same name. This means that you do not have to worry about how other users name their files.

The maximum length for a directory name is 16 characters, the maximum length for a full pathname is 44.

You use the CREATE\_DIR (CRDIR) command to create a directory. You enter the CRDIR command as follows:

```
CRDIR NAME=<master-directory-name>.<directory-name>;
```

**NAME** is the name of the master directory under which you want to create your directory, plus the name you want to give your directory.

The master directory is simply the first directory level in the tree structure. Generally, the master directory already exists.

The following is an example of creating a directory:

```
CRDIR NAME=DEV.TEAM3
```

Do not create too many levels of directories; limit yourself to one level below the master directory when possible. File management can become very complicated if your tree structure is too deep.

### 3.3.3 Listing Information about a Library

#### Listing a file

To get information about a library and its subfiles, you use the LIST\_FILE (LSF) command. You enter the LSF command as follows:

```
S: LSF FILE=<file-name> <parameters>;
```

FILE is the name of the file for which you want information.

The following is an example of listing information about the organization of the DEV.TEAM1 file. You enter:

```
S: LSF FILE=DEV.TEAM1.SL ORG=1;
```

## Developing Applications

The system displays the following screen (Figure 3-15):

```
DEV.TEAM1.SL                                OWNER : DEV

ORG
  format= LINKQD  catalog.= YES  blksize = 4070+26  expdate
=94/356
  organiz= LINKQD  recform = VB   incrsz= 0        unit   =BLK
  recsize= 264 +4  delrec  = VB   compact = NO      type   =SL
  dirsize= 5       maxsize = 0   logtrksz= 1
<<<13:23
S:
```

**Figure 3-15. Information Listed about a Library**

ORG is the parameter you must include for organization information.

## Getting Started With IOF

### Listing file space

The LIST\_FILE\_SPACE (LSFSP) command allows you to list the amount of space allocated to a particular file. You enter the LSFSP as follows:

```
S: LSFSP FILE=<file-name>;
```

FILE is the name of the file for which you want information.

Below is an example of listing file space:

```
LSFSP FILE=DEV.TEAM1.SL
```

The following is displayed on your screen (Figure 3-16):

```
DEV.TEAM1.SL                                OWNER : DEV

SPACE
  ON      EXT-SN    SIZE  UNIT    START_<-DBLK->_END_%USABLE
  MSDI01    0      250  BLK     602600    604599    100
<<<13:25
S:
```

**Figure 3-16. Listing File Space**

## Developing Applications

### Listing a directory

The LIST\_DIR (LSDIR) command lists a directory and the objects it contains. You enter the LSDIR command as follows:

```
S: LSDIR NAME=<directory-name>
```

NAME is the name of the directory you want to list.

If you do not include the name parameter, the current working directory is listed.

Below is an example of listing a directory using the SHORT parameter.

```
S: LSDIR DEV SHORT
```

SHORT when this parameter is used only the tree structure, without any complementary information, is listed.

## Getting Started With IOF

This screen, Figure 3-17 shows the type of the information the system displays when you execute the LSDIR command.

```
>>>13:30 LSDIR 30.00
      (FROM CATALOG: DEV.CATALOG ON MEDIA: MSDI01)

DEV -> master directory
DEV.CATALOG -> file

DEV.TEAM1 -> directory
DEV.TEAM1.BIN -> file
DEV.TEAM1.SL -> file

DEV.TEAM2 -> directory
DEV.TEAM2.SL -> file

DEV.TEAM3 -> directory

DEV.TEAM4 -> directory

      number of listed objects
<<<13:30
S:
```

**Figure 3-17. Example of Listing a Directory**

The system gives you the total number of objects in the directory and the time of the request, as well as the tree structure itself.



### 3.4 PRINTING LIBRARY MEMBERS

To print library members you must use the DPRINT (DPR) command. (To print libraries, use the PRINT\_FILE command.) The DPR command is entered as follows:

```
S: DPR SUBFILES=<subfile-names>  
      LIB=<library-name>
```

SUBFILES are the names of the subfiles you want to print.

LIB is the name of the library in which the subfiles you want to print reside.

For example, type:

```
S: DPR SUBFILES=JONES-STP  
      LIB=DEV.TEAM1.SL
```

and then press **ENTER**.

## Getting Started With IOF

The equivalent in Novice (Menu) mode produces the following screen, Figure 3-18.

```
1/1                DPRINT                (+)-->:___
                print subfiles on a remote printer
                JONES-STP
SUBFILES          subfiles to be printed
LIB              source library (default is #SLIB)
                DEV.TEAM1.SL
CLASS            class (A to Z)
PRIORITY         priority (0 to 7)
WHEN             IMMED, END
DEST            destination [host.]station
TWDEST          destination terminal/pool
NAME            name of the delivery
BANINF          banner information (up to 4 lines)
COPIES          number of copies                1
NUMBER          generate line numbers in delivery ? 0
-----
+++
```

**Figure 3-18. DPRINT Screen in Novice Mode**

Press **ENTER**.

The system then generally displays a message either acknowledging execution of the DPRINT command, or failure to execute. The message format depends on your system configuration.

### 3.5 DELETING LIBRARIES

There are two ways to delete a library:

- You can delete the contents of a library, and keep the library for further use.
- You can delete the whole library, and free its disk space.

The following are examples of the two ways to delete using the example libraries given above in Section 3.3.

To delete only the contents, use the CLEAR\_LIBRARY (CLRLIB) command. For example:

```
CLRLIB DEV.TEAM2.SL;
```

The system responds with:

```
FILE DEV.TEAM2.SL HAS BEEN CLEARED
```

Another CLRLIB example:

```
CLRLIB TEMPSL$TEMPRY;
```

With a system response of:

```
FILE 102441.TEMPSL HAS BEEN CLEARED
```

## Getting Started With IOF

To delete the whole library, use the DELETE\_LIBRARY (DLLIB) command.

```
DLLIB DEV.TEAM2.SL
```

The system responds with the following, Figure 3-19

```
SPACE DEALLOCATED FOR THE FILE / DEV.TEAM2.SL
      VOLUME :MSDI01 DEALLOCATION SUCCESSFUL
CATALOG WAS UPDATED
<<<13:45
S:
```

**Figure 3-19. Deleting a Whole Library**

## Developing Applications

### 3.6 USING GCL PROCEDURES

Up to this point, all the commands discussed in this manual have been standard GCL commands. IOF provides a facility to define your own commands. To do this, you must write a program that defines the new command. This program is called a GCL procedure.

#### 3.6.1 Creating a GCL Procedure

To create your GCL procedure, you must be in the MAINTAIN\_COMMAND processor.

To execute the MAINTAIN\_COMMAND (MNCMD) processor, you must enter:

```
S: MNCMD [BINLIB=library-name];
```

**BINLIB**            the identification of the binary library where compiled GCL procedures are stored and from which they are retrieved (by default, the working output binary library).

The system replies with a message similar to the following:

```
>>>10:46 MNCMD 30.00 21-1
```

You are now at the C: prompt (command level). At the C: prompt you must execute CR (CREATE). Type:

```
C: CR;
```

and then press **ENTER**.

## Getting Started With IOF

### Example GCL Procedure

We are going to create a GCL procedure named WELCOME which displays a welcome message, "GOOD MORNING", followed by the name of the individual. It will have a single keyword (KWD) called PARAM1. This keyword will be used to store the name of the individual.

You can see this procedure in Figure 3-21 and the results of its execution in Figure 3-22.

The following is a step by step explanation of the creation of this GCL procedure.

Starting at the system prompt (S:), as explained just above, you **ENTER**:

```
S:MNCMD BINLIB=DEV.TEAM1.BIN;
```

The system returns:

```
>>>13:54 MNCMD 30.00 21 -1  
C:
```

Execute the CREATE command:

```
C:CR;
```

The system now prompts you to enter the commands to create your GCL procedure.

On the first line (10:), you enter the PROC command, which defines the name of your procedure. This is followed by the name you want to give your procedure. Here the name is WELCOME. Thus, you type:

```
10:PROC WELCOME;
```

and then press **ENTER**.

## Developing Applications

The system gives the next line (20:). Here you **ENTER** a ? to get a menu of possible GCL commands.

The first page of a 2 page menu is displayed. Press **ENTER** to get the second page of the menu, which is the following screen (Figure 3-20).

```
2/2          GCL
1 KWD          define a keyword
2 LABEL        define a label
3 LET          assign a value to a variable
4 LOCAL        define a local variable
5 OTHERWISE    OTHER head not selected cases in case of
                block
6 PROC          head a procedure definition
7 REPLY        retry a current calling sequence
8 RETURN       return to caller
9 SCALL        scan and call a GCL procedure
10 SYSTEM      ask for execution of command
11 UNLIST      head a UNLIST block
12 UNTIL       head a UNTIL block
13 VCALL       verbose activation of a GCL procedure
14 VCHAIN      verbose transfer to a GCL procedure
15 WHILE       head a WHILE block

                -->:
:
:
:
```

**Figure 3-20. GCL Menu**

Select the KWD command from the menu to define the names of the parameters and their descriptions if there are any.

## Getting Started With IOF

```
1/2          KWD          (+)-->:___
          define a keyword

NAME          + name of keyword and its aliases
              PARAM1

TYPE          type of the variable          NAME
LENGTH        maximum length of data
NUMVAL        minimum and maximum number of elements

DEFAULT       default value

VALUES        one of the conditions must be met

PROMPT        prompting text

-----
+++
```

**Figure 3-21. Keyword Parameter Screen**

Here you enter PARAM1 and its type, TYPE=NAME. Thus, you can enter the values in the screen above (Figure 3-21) and then press **ENTER**.

Here we continue in line mode.

The system gives you the next line (30). Here you enter the LET # #CAT command. In this case the LET command is used to display a message on the console when the procedure is executed. It is followed by your message ('Good Morning Mr' %PARAM1), with its parameter. Thus, you type:

```
30:LET # #CAT('Good Morning Mr ' %PARAM1);
```

and then press **ENTER**.



## Developing Applications

The system gives the next line (40:). You have finished defining your procedure. To end the definition of a procedure you must use the ENDPROC command. Thus, you type:

```
40:ENDPROC;
```

and then press **ENTER**.

The system gives you the next line (50:). To leave the command CR , you must enter a slash (/). Thus you type:

```
50: /
```

and press **ENTER**.

You are back at C: (command level). Here you must declare the domain in which you want to put the command. The domain used in this example is IOF. Thus, you type:

```
C: domain IOF;
```

and then press **ENTER**.

You are back at C: again. You must enter the command SV (SAVE) to save your GCL procedure. Thus, you type:

```
C: SV;
```

and then press **ENTER**.

To exit the MAINTAIN\_COMMAND processor, you must enter a slash (/). Thus:

```
C: /
```

You are back at S: level.

## Getting Started With IOF

The following figure shows what the procedure described above looks like on screen (Figure 3-22).

```
S: mncmd
>>>13:54 MNCMD 30.00 21 -1
C: cr;
  10:PROC WELCOME;
  20:?
I: KWD PARAM1
TYPE=NAME;
  30:LET # #CAT('Good Morning Mr ' %PARAM1);
  40:ENDPROC;
  50:/
C: DOMAIN IOF;
C: sv;
C: /
<<<14:02
S:
```

**Figure 3-22. On Screen View of Creating GCL Procedure**

### 3.6.2 Executing a GCL Procedure

To execute a GCL procedure (GCL Command) in line mode, you must enter the procedure name followed by the parameters asked by the procedure.

The syntax for executing a GCL procedure is the following:

```
S: procedure name [parameter1 [parameter]]
```

The screen below (Figure 3-23) executes the GCL procedure detailed in the previous section of this chapter.

```
S: WELCOME JONES;  
Good Morning Mr Jones  
S:
```

**Figure 3-23. Example of Executing GCL Procedure**

Getting Started With IOF

## 4. More About Files and Libraries

### 4.1 CREATING LIBRARIES

The process of creating a library consists of giving it a name and a type, allocating space on a disk, and initializing its structures, in particular creating the directory of members.

All this is done with the BUILD\_LIBRARY (BLIB) command. This command is described in Chapter 3.3 of this manual, the following is an example to remind you about BLIB syntax.

The general form of the BUILD\_LIBRARY command is as follows:

```
BLIB LIB=<library-description>:media:dvc
      UNIT=<CYL | BLOCK | 100KB>
      FILESTAT=<CAT | UNCAT | TEMPRY>
      SIZE=<allocation-size>
      MEMBERS=<number-of-members>
      TYPE=<SL | CU | LM | SM | BIN>
      INCRSIZE=<increment-size>
      MAXSIZE=<maximum-size>
      EXPDATE=<expiration-date>
      COMPACT=bool
```

Most of these mandatory or optional parameters are self-explanatory. There are simple explanations for some of them in Section 3.3 of this manual.

## Getting Started With IOF

There are 5 different types of libraries:

- Source Language (SL)
- Compile Unit (CU)
- Load Module (LM)
- Sharable Module (SM)
- Binary Object (BIN)

These libraries can be briefly described as follows.

- Source Language libraries can hold any kind of textual information, including programs written in different languages.
- Compile Unit libraries can only contain the results of the compilation of source language programs.
- Load Module libraries contain executable programs as resulting from the processing of CUs by LINK\_PG.
- Sharable Module libraries contain executable TPRs (Transaction Processing Routines) to be executed under TDS (Transaction Driven Subsystem) as resulting from the processing of CUs by LINK\_PG.
- Binary Object libraries can contain any other kinds of binary (not legible by eye) objects; this includes GCL command definitions, DDL schemas, etc.

You cannot introduce a member into a library if that member is not of the same type as all the other items held in the library. For example, if you try to introduce a Compile Unit member into a Source Language library, the Compile Unit member is rejected. This mechanism is designed to protect you against errors, and is known as the "type" protection mechanism.

## More About Files and Libraries

The following are some examples of building different libraries:

1. BLIB DEV.TEAM2.CU:MSDI01:MS/FSA TYPE=CU  
SIZE=10 MEMBERS=200

Where you create a cataloged CU library of 10 cylinders, with a directory for 200 members, on the FSA volume MSDI01.

2. BLIB TEMPLM\$RES FILESTAT=TEMPRY TYPE=LM  
SIZE=1 MEMBERS=10

Where you create a temporary LM library on a resident volume.

3. BLIB DEV.TEAM2.BIN:MSDI01:MS/FSA UNCAT  
TYPE=BIN SIZE=12 MEMBERS=50 INCRSIZE=1  
MAXSIZE=20

Where you create an uncataloged Binary Object library with an increment and a maximum size.

## 4.2 CREATING CATALOGED FILES

To create a catalog a file, the catalog must exist. If necessary, you must first create a catalog.

### 4.2.1 Creating a Catalog

A catalog is a file which consists of a set of records arranged in a hierarchical tree structure.

There are three categories of catalogs:

- the site catalog
- the system catalog
- private catalogs

These categories can briefly be described as follows:

- The site catalog, called SITE.CATALOG, contains all information required for controlling access to the system (users' names, passwords, projects, billings, environments, applications, etc.).
- The system catalog, called SYS.CATALOG, contains entries describing system files (that is, files whose names begin with SYS) and is the only catalog that is allowed to contain such files.
- A private catalog contains objects (master directories, directories, files, file links, generation groups) related to one project.



## More About Files and Libraries

In this manual we are primarily concerned with private catalogs.

There are two types of private catalogs:

- private catalogs whose objects can be accessed without specifying in which catalog file they are to be searched for. These catalogs are called automatically attachable catalogs.
- private catalogs that must be explicitly connected (attached) to the session in order to access the objects that they contain.

It is preferable to use automatically attachable catalogs, then the user does not have to bother with the concept of catalogs. Use other catalogs only when migrations from site to site are frequent.

You use the command `CREATE_CATALOG` (CRCAT) to create an automatically attachable catalog.

The syntax of the CRCAT command is as follows:

```
CRCAT NAME=<catalog-description>  
      NBOBJECT=<number-of-objects>  
      VOLUME=<volume-description>  
      AUTOATT=<1/0>  
      EXPDATE=<expiry-date>  
      PRTRFILE=<report-file>
```

## Getting Started With IOF

NAME	the name of the catalog you want to create. If the suffix .CATALOG is not supplied, it is automatically assumed. If AUTOATT is 1 (default), the name of the catalog must be in the form X or X.CATALOG (that is, may not contain more than one period). The directories under which you want to create the catalog must already exist in the site catalog. For example, if you want to create catalog DEV.CATALOG, the master directory DEV must already exist in the Site Catalog.
NBOBJECT	is the maximum number of objects you want to store in the catalog. This value determines the amount of space you allocate for the catalog; the maximum value is 100000.
VOLUME	is the volume on which you want to create the catalog. This must be a disk volume.
AUTOATT	if 1 (default), the catalog is automatically attachable. In this case Name must specify a project name only, or a project name plus suffix .CATALOG. If 0, the catalog is not automatically attachable. Use automatically attachable catalogs.
EXPDATE	you express the expiration date of the catalog as follows:  nnn            a number of days yy/ddd        day in the year yy/mm/dd      day in the month and year  the default is no expiration date (that is, today's date).

## More About Files and Libraries

**PRTFILE**            the report file. When you omit it, the report is displayed by default at the user's terminal.

**Note:**            When AUTOATT is 1 (default), NAME must be in form X or X .CATALOG, where X is a project name. The following is an example of creating a catalog.

```
CREATE_CATALOG DEV NBOBJECT=300
VOLUME=MSDI01:MS/FSA
```

Once you have created this catalog, the system knows to create or retrieve all cataloged objects whose names begin with DEV, in DEV.CATALOG. The only restriction is that the master directory DEV must already exist in the Site Catalog. You create directories by means of the CREATE\_DIR command. This command is discussed in section 3.3.2 of this manual.

Once the directory exists, commands like:

```
BUILD_LIBRARY            DEV.TEAM1.SL
LIST_FILE                DEV.TEAM1.SL
EXEC_PG MYPG             FILE1=IFN
ASG1=DEV.TEAM1.SL
etc.
```

refer to the catalog DEV.CATALOG.

## Getting Started With IOF

Occasionally, the system cannot find the catalog from which you require it to retrieve objects. This is very rare, and occurs:

- when creating a new master directory within the private catalog DEV.CATALOG.
- when setting access rights for the whole catalog starting from the catalog root.
- when manipulating a fileset, and the name of the master directory does not appear in the fileset, for example: \*.AB or \*\*, etc.,

For more information about catalog management see the *Catalog Management User's Guide*.

For more information about fileset syntax see the *IOF Terminal User's Reference Manuals*.

In all these cases, you must specify the catalog name. For example:

```
LIST_FILESET  **$CAT:DEV.CATALOG
CREATE_DIR    DEV-BIS CATNAME=DEV.CATALOG
MODIFY_ACL   * DEV/READ CATNAME=DEV.CATALOG
```

In this example, once the directory is created, all names that begin with DEV-BIS are searched for in the private catalog DEV.CATALOG.

#### 4.2.2 Creating a Cataloged File

##### **Catalog structure**

As mentioned above, a private catalog consists of a set of records arranged in a hierarchical tree structure. Each record describes a cataloged object, identified by a unique object name.

The objects that can be cataloged are listed here in descending hierarchical order.

- master directories
- directories
- files (including catalog files)

Each object is cataloged in a tree structure that descending from a common root, which has no name.

The rules listed below apply when you catalog an object.

- Master directories can only be cataloged under the root.
- Directories can only be cataloged under a master directory or another directory.
- A file can only be cataloged under a directory.

Here, we are interested in cataloging files.

A cataloged file is a permanent file whose identification is held in a dedicated file, a catalog. Once a file is cataloged, you need specify only its name to reference it. The system retrieves from the catalog the information relative to the volume(s) on which the file resides.

You can protect cataloged files by associating a list of access rights with them, thus preventing unauthorized access.

## Getting Started With IOF

As mentioned in the previous section, several other attributes can be kept in the catalog. These include the authorized level of sharing (for example, concurrent access by several users), the type of file integrity, etc., together with information such as the last update date, last file version, current generation number, etc.

### How to create a cataloged file

Now that you have a catalog, you can create a cataloged file.

One way to create a cataloged file is to use the BUILD\_FILE (BF) command.

The following is an example of creating a cataloged file on a volume named MYVOL using the BUILD\_FILE command:

```
BF A.B.C:MYVOL:MS/FSA UFAS=INDEXED UNIT=100KB
  SIZE=5  CISIZE=1024 RECSIZE=100 RECFORM=V
  KEYLOC=1 KEYSIZE=10 CIFSP=20
```

In this example the parameters represent the following:

UFAS	the format and organization of the file, here it is UFAS INDEXED.
UNIT	the unit of allocation the utility uses, here it is 100KB.
SIZE	when UNIT is expressed as 100KB, SIZE specifies the total amount of space to be allocated to the file expressed as a number of 100KBs.
CISIZE	the Control Interval size for the file. The value supplied is in units of bytes, here, 1024.
RECSIZE	the logical record in bytes, here, 100.

## More About Files and Libraries

RECFORM	the record format. It can be fixed (F) or variable (V). Here it is V.
KEYLOC	the location of the primary within the record, expressed as the position of its leftmost byte. In this example, 1.
KEYSIZE	the length of the primary key, expressed as a number of bytes. Here, 10.
CIFSP	the free space to be left in each CI when the file is sequentially loaded (opened in output). Given as a percentage, here 20.

## Getting Started With IOF

### 4.2.3 Accessing a Cataloged File

#### Cataloged file syntaxes

The syntax for specifying a cataloged file is as follows:

```
path-name [/g-suffix] [... subfile] [$CAT[i]]
```

**path-name** is a full path name, relative ascending, or relative descending path name. The simple names that build up a path name cannot exceed 16 characters in length. The characters allowed are the alphabets (A through Z), the numerics (0 through 9), the dash (-), and the underscore (\_).

**g-suffix** is the optional generation and/or version identification.

**subfile** is the name of a subfile (member) in a library which is to be treated as a sequential file. The name must not exceed 31 characters in length.

**\$CAT** is an optional suffix indicating that the file is cataloged.

**i** is the rank number of the in the ATTACH statement (when used).

#### Examples of cataloged file names.

The full path name of a cataloged file:

```
DEV.TEAM1.REF
```



## More About Files and Libraries

The full path name of a cataloged file:

```
DEV . TEAM1 . SL $CAT
```

Member JONES-STP of library DEV.TEAM1.SL

```
DEV . TEAM1 . SL . . JONES - STP
```

A relative descending path name:

```
. SL
```

A relative ascending path name:

```
<TEAM1 . SL
```

Getting Started With IOF

## Glossary

### **billing**

An accounting concept; all resources consumed by a set of users can be charged to a single billing.

### **bin library**

A library that contains miscellaneous binary objects such as command definitions, form definitions, etc.

### **boolean**

One of the types that can be associated with a GCL variable; a boolean variable can take values 0 or 1 to denote respectively no or yes.

### **break**

A signal that is used to interrupt processing; this signal can be sent by a dedicated key on the terminal or by the `*$BRK` local network command.

### **catalog**

A special permanent file that contains protection information and user-associated information for files, volumes, etc.

## Getting Started With IOF

### **cataloged file**

A file whose name and attributes are stored in another file called a catalog. (Most permanent files are cataloged.)

### **character**

One of the types that can be associated with a GCL variable; a character variable can contain up to 255 characters; any character may be used.

### **command**

A statement of the command language.

### **command language**

A set of rules and conventions to be used for expressing an action to be performed in a way that is acceptable to a computer; GCL is the GCOS 7 command language.

### **compiler**

A processor that generates a Compile Unit from a source program; a source program is stored in an SL library; a Compile Unit is stored in a CU library.

### **CU library**

A library that contains Compile Units; a Compile Unit results from the processing of a source program in an SL library by a compiler.

## Glossary

### **device class**

The type of disk or tape cartridge that describes the volume.

### **directive**

A command that can be issued at any time when the system requests input, as opposed to other commands that can be entered only in specific contexts.

### **directory (file system)**

A mechanism that provides independence between users as far as file naming is concerned; a catalog is arranged as a tree-structured hierarchy of directories.

### **directory (of a library)**

The part of a library that contains the names of the members and is used to access them.

### **Distributed System Architecture (DSA)**

A set of architectural concepts, protocols, rules, and facilities whose common aim is to allow terminals, computers, and switch nodes, possibly of different types, to communicate and cooperate; fundamental to DSA are the concepts of layered structure and decentralization.

### **environment**

A set of related commands made accessible to users of a project to carry out a predefined type of task.

## Getting Started With IOF

### **file**

One of the types that can be associated with a GCL variable; a file variable refers to a container for records of information; it can be read from or written into.

### **Front End Processor (FEP)**

A piece of equipment that executes the DNS (Distributed Network Supervisor) to control communication with several host systems, possibly of different types.

### **Full Screen Editor (FSE)**

The GCOS7 text editor operating on full screens.

### **GCOS 7**

General Comprehensive Operating System on the DPS 7000.

### **GCL**

The GCOS 7 command language.

### **help text**

An informative text that is displayed on request to assist the user.

### **hexadecimal**

One of the types that can be associated with a GCL variable; a hexadecimal variable can contain 2, 4, 6, or 8 hexadecimal digits; hexadecimal digits are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

## Glossary

### **host**

In the DSA context, refers to a connected system which is capable of executing jobs; DPS 6000, DPS 7000, and DPS 8000 are hosts, whereas terminals and front-end processors are not hosts.

### **Interactive Operating Facility (IOF)**

The GCOS 7 time-sharing application.

### **input library**

A library used to read information from.

### **job**

An execution unit on a host system.

### **library**

One of the types that can be associated with a GCL variable; a library is a particular type of file that can contain objects of various types (these are known as library members).

### **line mode**

A method of communicating with GCOS 7 in which the user enters the commands line by line, that is, without menus.

### **LM library**

A library that contains Load Modules; Load Modules result from the processing of Compile Units in a CU library by LINK\_PG.

## Getting Started With IOF

### **log-on**

The process for gaining access to a particular host system.

### **mailbox**

A container for messages sent to a user by other users or by the system.

### **member**

A unit in a library.

### **menu**

A device which allows the selection of one among a set of proposed options.

### **menu mode**

A method of communicating with GCOS 7 in which the user is guided by the use of menus.

### **network**

A set of interconnected hosts, sites, FEPs, switches, and terminals.

### **object**

One of the things upon which commands operate; files, libraries, catalogs, directories, programs, variables, commands, etc. are objects.



## Glossary

### **output library**

A library used for read and write operations.

### **password**

A string of up to 12 characters long, used to allow a specific user access to the system.

### **permanent file**

A file created for long-term use and stored on a volume.

### **private catalog**

A catalog that holds information on the files belonging to one project only.

### **procedure (GCL)**

A sequence of GCL commands which defines a new command.

### **processor**

A group of commands that performs a specific set of functions in GCOS 7.

### **program**

A sequence of instructions that can be understood by a computer.

## Getting Started With IOF

### **project**

A group of logical users.

### **prompt**

A request from the system to supply one or more values.

### **SL library**

A library that contains Source Language objects (that is, objects that can be read by a human being).

### **SM library**

A library that contains Sharable Modules; Sharable Modules result from the processing of Compile Units in a CU library by LINK\_PG; Sharable Modules are used for holding TPRs to be executed under the Transaction Driven Subsystem (TDS).

### **startup sequence**

A registered sequence of GCL commands or other input data which is automatically read in at log-on time.

### **station**

(also known as logical station) A local image of the systems in the network, regardless of the actual physical configurations of sites and hosts; a logical station is defined in the system catalog by the System Administrator, as being one or more sites; a given site may belong to several logical stations.

## Glossary

### **system administrator**

A privileged user who is in charge of defining users, projects, and billings in the catalog, who sets and controls access rights to various objects, who installs the system and tailors the system files, etc.

### **system variable**

A variable with a predefined name used to define an element of the user profile.

### **type (of a library)**

An attribute associated with a library that defines the kind of the members that it can contain; library types are SL, CU, LM, SM, and BIN.

### **type (of a variable)**

An attribute associated with a variable that defines the values that it can take; types of variables are: boolean, character, decimal, file, fileset, hexadecimal, library, name, output, remote file, star-name, and volume.

### **uncataloged file**

A file whose name and attributes are not stored in another file called a catalog. (Temporary files are never cataloged, most permanent files are cataloged.)

### **user name**

A name that identifies the terminal user within the network.

## Getting Started With IOF

### **user profile**

A set of system variables that determine how a particular user is to be handled by the system.

### **variable**

A symbolic name for a user-defined or system-defined value that can change in the course of an IOF session.

### **Volset**

A logical grouping of one or more disk volumes registered under a unique name in the Storage Administration file (SITE.QUOTA). When the VOLSET facility is active (that is, validated by the System Administrator), users do not need to give any location details when allocating a cataloged or temporary file. The file is automatically allocated on a volume in the Site Volset or, if it exists, the project's default Volset.

### **volume**

One of the types that can be associated with a GCL variable; a volume variable is used to denote volumes of many types (disks, tapes, cartridges, diskettes, etc.).

### **working directory**

The directory that contains a user's working files.

## Index

### A

accessing a cataloged file 4-12  
allocating library space 3-26  
APPEND command 3-18

### B

billing 1-11, g-1  
BIN library g-1  
Binary Object libraries 3-1, 4-2  
boolean type g-1  
break g-1  
BUILD\_FILE command 4-10  
BUILD\_LIBRARY command 3-23, 4-1  
building libraries  
    examples 4-3  
bypassing startups 2-7

### C

calling FSE 3-6  
catalog structure 4-9  
cataloged file syntax 4-12  
cataloged files 1-5, 1-6  
    accessing 4-12  
    creating 4-4, 4-9, 4-10  
    definition g-2  
    examples of names 4-12  
    names 4-12  
    syntax 4-12

## Getting Started With IOF

catalogs	1-3, 4-4
categories	4-4
creating	4-4
definition	g-1
private	4-4, 4-5
private (definition)	g-7
site	4-4
system	4-4
character type	g-2
CLEAR_LIBRARY command	3-35
command	g-2
BUILD_FILE	4-10
BUILD_LIBRARY	4-1
CREATE_CATALOG	4-5
CREATE_DIR	4-7
command language	g-2
commands	1-9
APPEND	3-18
BUILD_LIBRARY	3-23
CLEAR_LIBRARY	3-35
CREATE	3-10, 3-38
CREATE_DIR	3-27
DELETE_LIBRARY	3-36
DISPLAY_PROFILE	2-26
DPRINT	3-33, 3-34
ENDPROC	3-41
executing	2-17, 2-30
GCL	1-9
KWD	3-39
LIST_DIR	3-31
LIST_FILE	3-28
LIST_FILE_SPACE	3-30
MODIFY	3-14
MODIFY_PROFILE	2-26, 3-12
MPRTLIB	3-4, 3-13
MWINLIB	3-1, 3-2
MWLIB	3-1, 3-3
PRINT_FILE	3-33
PROC	3-38
RENUMBER	3-20, 3-21
Compile Unit	3-1, 4-2
compiler	g-2
Connecting to IOF	2-1

## Index

CREATE command	3-10, 3-38
CREATE_CATALOG command	4-5
CREATE_DIR command	3-27, 4-7
creating	
a catalog	4-4
a directory	3-27
a GCL procedure	3-37
a subfile or member	3-10
an SL library	3-23
an SL library(examples)	3-25
cataloged files	4-4, 4-9, 4-10
cataloged files (example)	4-10
libraries	4-1
CU library	g-2

## D

default	
libraries	3-1
libraries (selecting)	3-1
options (logging on)	2-4
values	2-19
DELETE_LIBRARY command	3-36
deleting libraries	3-35
device class	g-3
difficulties	2-10
directives	1-9, g-3
directories	1-3
creating	3-27
file systems (definition)	g-3
libraries (definition)	g-3
working (definition)	g-10
DISPLAY_PROFILE command	2-26
Distributed System Architecture (DSA)	g-3
DPRINT command	3-33, 3-34
DPS 7000	1-12
DSA	g-3

## Getting Started With IOF

### E

EDIT	1-7, 3-5
ENDPROC command	3-41
environment	g-3
error messages	2-10
example	
building libraries	4-3
getting help	2-23
menu	2-29
executing	
a GCL procedure	3-43
commands	2-17, 2-30

### F

FEP	g-4
files	1-3
cataloged	1-5, 1-6
cataloged (definition)	g-2
definition	g-4
permanent	1-5
permanent (definition)	g-7
temporary	1-5
uncataloged	1-5, 1-6
uncataloged (definition)	g-9
files, directories and catalogs	1-3
Front End Processor (FEP)	g-4
FSE	1-7, 3-5, 3-6, 3-10
choosing an option	3-9
definition	g-4
Full Screen Editor (FSE)	g-4



## Index

### G

GCL	1-9, 3-37, 3-38, 3-43
commands	1-9
creating a procedure	3-37
definition	g-4
directives	1-9
executing a procedure	3-43
procedure	g-7
procedure (example)	3-38
procedures	3-37
GCOS 7	g-4

### H

help	1-8, 2-21
example of getting	2-23
screens	2-21
texts	1-8, 2-21
texts (definition)	g-4
hexadecimal type	g-4
host (DSA)	g-5

### I

input libraries	3-1, 3-2
definition	g-5
Interactive Operating Facility (IOF)	g-5
IOF	
connecting	2-1
definition	g-5
DPS 7000	1-12
general information	1-10
key concepts	1-3
modes	1-1
session outline	2-28
what is it?	1-1

## Getting Started With IOF

### J

job g-5

### K

KWD command 3-39

### L

libraries 1-3, 3-1  
  Binary Object 3-1, 4-2  
  Compile Unit 3-1, 4-2  
  creating 4-1  
  definition g-5  
  deleting 3-35  
  input 3-1, 3-2  
  input (definition) g-5  
  Load Module 3-1, 4-2  
  output 3-1, 3-3  
  output (definition) g-7  
  Sharable Module 3-1, 4-2  
  Source Language 3-1, 3-2, 4-2  
  types 4-2  
libraries and members 1-3  
line mode 1-1, 3-9, g-5  
LIST\_DIR command 3-31  
LIST\_FILE command 3-28  
LIST\_FILE\_SPACE command 3-30  
listing output libraries 3-4  
LM library g-5  
Load Module 3-1, 4-2  
log-on g-6  
logging off 2-27, 2-30

## Index

logging on	2-1
default options	2-4
IOF session	2-28
network software	2-2
specified options	2-6

## M

mailbox	g-6
mailboxes	2-8
MAINTAIN_COMMAND processor	3-37
exiting	3-41
MAINTAIN_LIBRARYprocessor	3-5
members	1-3
creating	3-10
definition	g-6
modifying	3-14
menu	
mode	g-6
menus	1-8
definition	g-6
screen description	2-15
using	2-14
modes	
line	1-1
menu	g-6
novice	1-1, 3-6
MODIFY command	3-14
MODIFY_PROFILE command	2-26, 3-12
modifying a subfile or member	3-14
MPRTLIB command	3-4, 3-13
MWINLIB command	3-1, 3-2
MWLIB command	3-1, 3-3

## Getting Started With IOF

### N

network	
definition	g-6
logging on	2-2
novice mode	1-1, 3-6

### O

object	g-6
options	
logging on	2-6
output libraries	3-1, 3-3
definition	g-7
listing	3-4

### P

parameters	
mandatory	2-19
optional	2-19
supplying values	2-19
supplying values (example)	2-20
passwords	1-10
definition	g-7
permanent files	1-5
PRINT_FILE command	3-33
printing library members	3-33
private catalogs	4-4, 4-5
definition	g-7
PROC command	3-38
Procedure (GCL)	g-7
processor	g-7
profiles	
user	1-2, 1-8
program	g-7

## Index

projects 1-11  
    definition g-8  
prompt g-8

## R

RENUMBER command 3-20, 3-21  
restarting an interrupted session 2-8

## S

selecting default libraries 3-1  
Sharable Module libraries 3-1, 4-2  
site catalogs 4-4  
SL libraries  
    creating 3-23  
    definition g-8  
SM library g-8  
Source Language libraries 3-1, 3-2, 4-2  
source text editors 1-7  
    EDIT 1-7  
    FSE 1-7  
startups  
    bypassing 2-7  
    definition g-8  
    using 2-11  
station g-8  
subfile  
    creating 3-10  
    modifying 3-14  
supplying parameter values 2-20  
system  
    administrator g-9  
    catalogs 4-4  
    variable g-9

## Getting Started With IOF

### T

temporary files	1-5
tree structure	1-4
type	
libraries	g-9
variables)	g-9

### U

uncataloged files	1-5, 1-6
definition	g-9
user	
names	1-10, g-9
profiles	1-2, 1-8, 2-25, g-10
using	
FSE	3-5
GCL procedures	3-37
menus	2-14
startups	2-11

### V

variable	g-10
Volset	3-23, g-10
volume type	g-10

### W

working directory	g-10
-------------------	------

## Technical publication remarks form

<b>Title :</b>	DPS7000/XTA NOVASCALE 7000 Getting Started With IOF Job Control and IOF
----------------	---

<b>Reference N° :</b>	47 A2 35UJ 00
-----------------------	---------------

<b>Date :</b>	January 1995
---------------	--------------

### ERRORS IN PUBLICATION

--

### SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

--

Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.  
If you require a written reply, please include your complete mailing address below.

NAME : \_\_\_\_\_ Date : \_\_\_\_\_

COMPANY : \_\_\_\_\_

ADDRESS : \_\_\_\_\_

Please give this technical publication remarks form to your BULL representative or mail to:

Bull - Documentation Dept.  
1 Rue de Provence  
BP 208  
38432 ECHIROLLES CEDEX  
FRANCE  
info@frec.bull.fr

# Technical publications ordering form

To order additional publications, please fill in a copy of this form and send it via mail to:

**BULL CEDOC**  
**357 AVENUE PATTON**  
**B.P.20845**  
**49008 ANGERS CEDEX 01**  
**FRANCE**

**Phone:** +33 (0) 2 41 73 72 66  
**FAX:** +33 (0) 2 41 73 70 66  
**E-Mail:** [srv.Duplicopy@bull.net](mailto:srv.Duplicopy@bull.net)

CEDOC Reference #	Designation	Qty
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
[ ] : The latest revision will be provided if no revision number is given.		

NAME: \_\_\_\_\_ Date: \_\_\_\_\_

COMPANY: \_\_\_\_\_

ADDRESS: \_\_\_\_\_

PHONE: \_\_\_\_\_ FAX: \_\_\_\_\_

E-MAIL: \_\_\_\_\_

**For Bull Subsidiaries:**

Identification: \_\_\_\_\_

**For Bull Affiliated Customers:**

Customer Code: \_\_\_\_\_

**For Bull Internal Customers:**

Budgetary Section: \_\_\_\_\_

**For Others: Please ask your Bull representative.**





**BULL CEDOC**  
**357 AVENUE PATTON**  
**B.P.20845**  
**49008 ANGERS CEDEX 01**  
**FRANCE**

REFERENCE  
**47 A2 35UJ 00**