

INTERCOMM

DYNAMIC FILE ALLOCATION



**ISOGON
CORPORATION**

330 Seventh Avenue, New York, New York 10001

LICENSE: INTERCOMM TELEPROCESSING MONITOR

Copyright (c) 2005, 2022, Tetragon LLC

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Use or redistribution in any form, including derivative works, must be for non-commercial purposes only.
2. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
3. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Dynamic File Allocation

Publishing History

<u>Publication</u>	<u>Date</u>	<u>Remarks</u>
First Edition	January 1977	This manual corresponds to Intercomm Release 7.0.
Second Edition	November 1982	Completely revised, updated and reorganized. This manual corresponds to Intercomm Release 9.0.
Third Edition	July 1987	Completely revised and updated. This manual corresponds to Intercomm Release 10.0.

The material in this document is proprietary and confidential. Any reproduction of this material without the written permission of Isogon Corporation is prohibited.

PREFACE

Intercomm is a state-of-the-art teleprocessing monitor system executing on the IBM System/370 family of computers and operating under the control of IBM Operating Systems (MVS and XA). Intercomm monitors the transmission of messages to and from terminals, concurrent message processing, centralized access to I/O files, and the routine utility operations of editing input messages and formatting output messages, as required.

This manual documents the use of the Dynamic File Allocation (DFA) Special Feature. DFA provides the capability to create new sequential data sets or access existing sequential or VSAM data sets without having to define the data sets explicitly via JCL.

This manual is intended for applications and systems programmers. The reader is assumed to be familiar with Intercomm.

INTERCOMM PUBLICATIONS

GENERAL INFORMATION MANUALS

Concepts and Facilities

Planning Guide

APPLICATION PROGRAMMERS MANUALS

Assembler Language Programmers Guide

COBOL Programmers Guide

PL/1 Programmers Guide

SYSTEM PROGRAMMERS MANUALS

Basic System Macros

BTAM Terminal Support Guide

Installation Guide

Messages and Codes

Operating Reference Manual

System Control Commands

CUSTOMER INFORMATION MANUALS

Customer Education Course Catalog

Technical Information Bulletins

User Contributed Program Description

FEATURE IMPLEMENTATION MANUALS

Autogen Facility

ASMF Users Guide

DBMS Users Guide

Data entry installation Guide

Data Entry Terminal Operators Guide

Dynamic Data Queuing Facility

Dynamic File Allocation

Extended Security System

File Recovery Users Guide

Generalized Front End Facility

Message Mapping Utilities

Model System Generator

Multiregion Support Facility

Page Facility

Store/Fetch Facility

SNA Terminal Support Guide

TCAM Support Users Guide

Utilities Users Guide

EXTERNAL FEATURES MANUALS

SNA LU6.2 Support Guide

TABLE OF CONTENTS

		<u>Page</u>
Chapter 1	INTRODUCTION	1
1.1	ALLOCATE Service Routine	1
1.2	ACCESS Service Routines	1
Chapter 2	USING THE SERVICE ROUTINES	3
2.1	Application Design Considerations	3
2.2	Parameters	4
2.3	Calls to ALLOCATE and ACCESS	4
2.4	Return Codes	6
2.5	Deleting Data Sets	6
Chapter 3	JCL REQUIREMENTS AND INSTALLATION PROCEDURES	7
3.1	JCL Requirements	7
3.2	Installation Procedures	7
Appendix A	CALLS AND RETURN CODES	9
A.1	Calling Syntax	10
A.2	DFACB Specifications and Return Values	11
A.3	Option Codes	14
A.3.1	ALLOCATE	14
A.3.2	ACCESS	15
A.4	Return Codes	15
A.4.1	ALLOCATE Return Codes in DDSASTAT	16
A.4.2	ACCESS Return Codes in DDSASTAT	17
Appendix B	DDSASECT DSECT	19
Index		21



List of Figures

	<u>Page</u>
Figure 1	File Handler Service Routines Used after ALLOCATE and ACCESS 2
Figure 2	Using DFACB for Calls to ALLOCATE or ACCESS 5



Chapter 1

INTRODUCTION

Dynamic File Allocation (DFA) provides a means by which programs (either Intercomm subsystems or batch programs) utilizing the File Handler may access already existing data sets or create new data sets without these data sets being explicitly defined via JCL. For non-catalogued data sets, it is only required that a DD statement be present in the Intercomm execution JCL which defines the disk pack containing (or to contain) the data set. Sequential and catalogued VSAM data sets residing on disk are supported. DFA functions are provided by two Dynamic File Allocation service routines: ALLOCATE and ACCESS. Explicitly defined (via JCL) data sets may be allocated and deallocated via the FILE command.

DFA routines are called by the application program in place of the normal call to SELECT. After a successful call to either ALLOCATE or ACCESS, the application programs may call the File Handler service routines (GET, PUT, GETV, PUTV, READ, WRITE, RELEASE) in the normal manner as illustrated in Figure 1. See also the appropriate Intercomm Programmers Guides.

1.1 ALLOCATE SERVICE ROUTINE

ALLOCATE allows application programs to create sequential (SAM) data sets in any format (fixed, variable or undefined) on any disk pack for which a DD statement is present. Dynamically allocated data sets have an implied disposition of NEW, and processing of them may subsequently be made only by calls to PUT or WRITE; calls to GET or READ are invalid. Dynamically allocated data sets may be catalogued at the time they are created. ALLOCATE cannot be used to create VSAM data sets.

Dynamically allocated data sets can be spun off for immediate use by other programs, such as batch print programs, or can be used as input to assemblers or compilers. Data sets created using ALLOCATE are not dependent on Intercomm once created (implied disposition is SHR), and can be used as any data set can be used. However, the programs that use dynamically allocated or accessed data sets require some caution because such data sets are not subject to the exclusive control normally provided by the operating system.

1.2 ACCESS SERVICE ROUTINE

ACCESS provides the ability to access existing SAM or VSAM data sets without having to explicitly declare in advance via JCL which data sets are to be accessed. It allows, for instance, a subsystem to

retrieve data from any number of SAM or VSAM data sets upon request by a terminal operator. Only the data set name (DSN) must be known. These data sets need not have been created prior to the execution of Intercomm; they must have been created at any time prior to being accessed. For VSAM, only existing, catalogued data sets can be accessed; ALLOCATE cannot be used to create VSAM data sets. In addition to being accessed, data sets may be extended or updated. Dynamically accessed data sets may have a disposition of OLD, (which allows retrieval or updating), a disposition of MOD (which allows extension of the data set) or a disposition of SHR (for VSAM data sets). VSAM data sets cannot have a disposition of MOD, and cannot be connected to an LSR pool. If a VSAM alternate index or path data set is allocated, the base cluster is also automatically allocated unless already defined in the execution JCL. In the latter case, updates should not be done because a GETV may not return the latest version of a record.

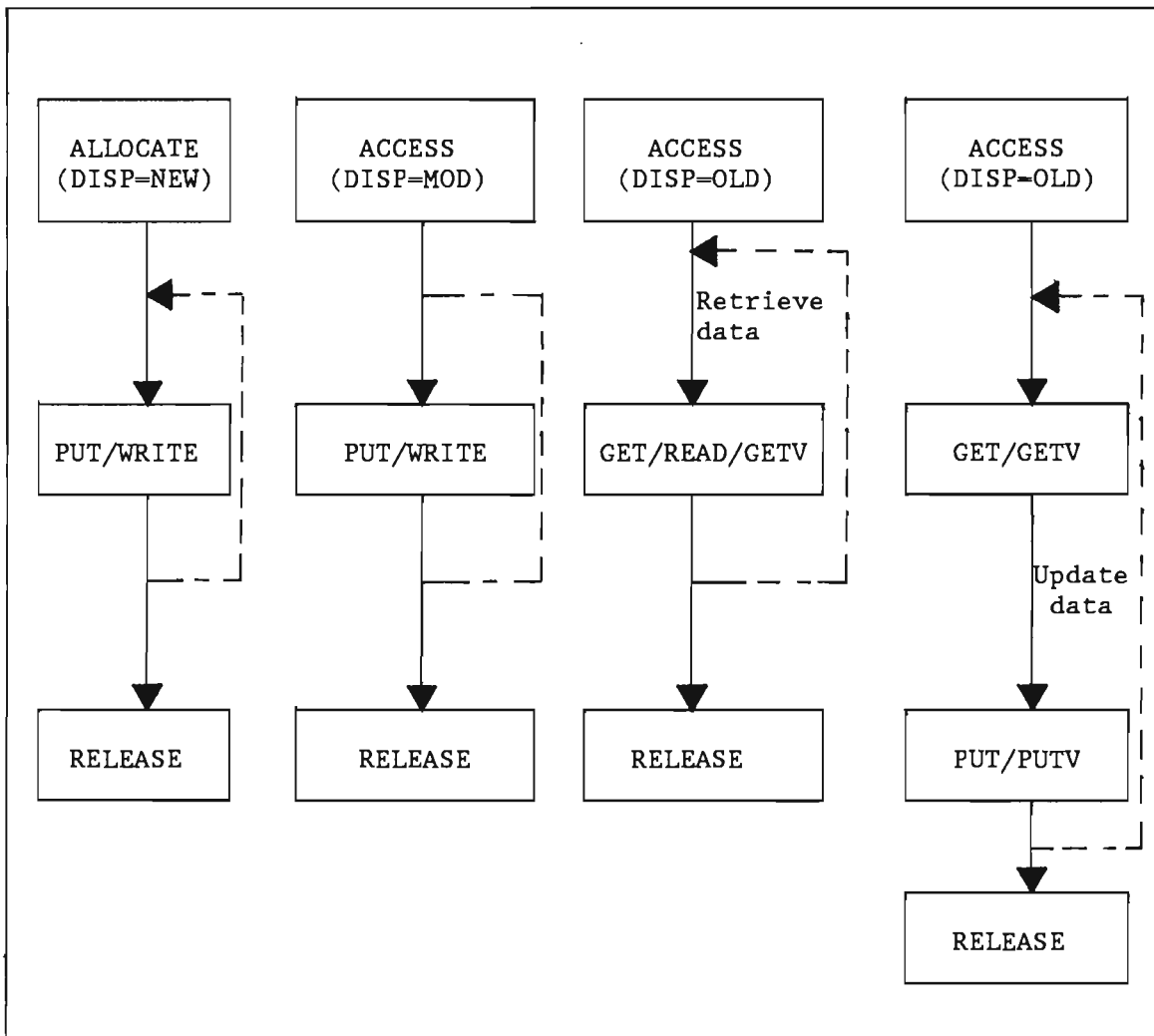


Figure 1. File Handler Service Routines Used After ALLOCATE and ACCESS

USING THE SERVICE ROUTINES

2.1 APPLICATION DESIGN CONSIDERATIONS

For SAM files, sufficient disk space must be available to contain the data to be written. The File Handler provides protection from an out-of-space condition (x37 abend) only for calls to WRITE (FHCW return code=2). The primary space allocation should be sufficient, by itself, to contain the anticipated data; no secondary quantity should be specified, since secondary space, if required, may or may not be available. If a secondary space is required but unavailable, an abend may result.

Volumes to contain allocated SAM data sets should be referenced by using the ddname rather than the volume-serial since this allows the physical volumes used for this purpose to be changed without program modification. If possible, after ALLOCATE is used to create a data set, all data should be written before RELEASE is called, because release processing automatically closes the data set and deallocates it from Intercomm. A data set previously allocated and released may be extended using ACCESS and specifying an option for a disposition of MOD. However, this involves additional OPENS and CLOSEs, and also reallocation and deallocation of the data set for every paired call to access and to subsequently release it.

Dynamically accessed data sets are opened in conjunction with the first call to GET, GETV, PUT, PUTV, READ or WRITE after calling ACCESS. Positioning is performed at the time the OPEN is issued. The data set is automatically closed when RELEASE is called. If it is desired that a SAM data set be deleted after processing, this can be accomplished in conjunction with RELEASE processing (see Section 2.5). A VSAM data set cannot be deleted.

An application need not have defined in advance the data set attributes of the files which it is to access. This information is returned after a call to ACCESS. RECFM indicates record structure and, therefore, processing logic; LRECL and BLKSIZE may be used to determine the size of required record areas. For a VSAM data set the CINV (control interval size) is returned instead of BLKSIZE.

Dynamically allocated or accessed data sets are not subject to the normal exclusive control provided by the operating system. Therefore, care should be used if a particular data set is to be concurrently accessed by multiple programs (on-line or batch) or by multiple threads of a reentrant program within Intercomm. One program can update a dynamically allocated data set while another program is concurrently reading that same data set. If this type of situation is not excluded by system design, the user may prevent concurrent accesses by enqueueing on the data set name. The Intercomm Enqueue/Dequeue facility should be used, as described in the Operating Reference Manual.

2.2 PARAMETERS

The parameters used with calls to ALLOCATE and ACCESS consist of the normal 12-fullword File Handler External Data Set Control Table area (EXTDSCT), the normal fullword File Handler Control Word (FHCW) and an additional 22-fullword Dynamic File Allocation Control Block (DFACB).

Except for data sets already catalogued, the user is required to provide in the DFACB an indication of the disk volume that is to contain the data set being allocated or that does contain the data set being accessed. The preferred method for ALLOCATE is to provide the eight-byte ddname of a DD statement which defines the disk volume. The alternative is to provide the actual volume-serial of the disk volume to contain (or containing) the data set. Calls to ACCESS for VSAM data sets should provide the data set name only.

Appendix B shows the Assembler Language definition of the DFACB that is supplied to the File Handler when application programs call ALLOCATE or ACCESS. Application programs coded in a high-level language must define this area as applicable to the language in use.

Appendix A.1 gives the specifications for calls to ALLOCATE and ACCESS, which are coded according to the standard Intercomm coding conventions for the language used. Appendix A.2 gives the specifications and return values of the fields of the DFACB. Appendix A.3 defines the option codes for the calls. Figure 2 shows the procedure for the calls.

2.3 CALLS TO ALLOCATE AND ACCESS

When calling ALLOCATE to create SAM data sets, the application program must supply the data set characteristics (RECFM, LRECL, BLKSIZE) as well as its space requirements. (Alternatively, this space information may be supplied on a DD statement. See Appendix A.2.) The application program must identify the disk pack to contain the data set, either by supplying the volume-serial or by supplying the ddname of a DD statement in the Intercomm execution JCL which references the proper disk pack. Cataloging of the new data set is requested via an option code.

When calling ACCESS, the data set name must always be supplied. If the data set is cataloged (required for VSAM), no other DFACB parameters are required. If the data set is not cataloged, the caller must supply either the volume-serial number of the disk pack which contains a SAM data set or the ddname of a DD statement which references the proper disk pack. (See Chapter 3.) Additionally, the caller may indicate that a SAM data set is to be extended by specifying a disposition of MOD via an option code. An empty VSAM-ESDS can be loaded after a call to ACCESS if the data set is predefined and cataloged via IDCAMS.

Note that use of a ddname, rather than a volume-serial, is recommended for both ALLOCATE and ACCESS for non-catalogued data sets.

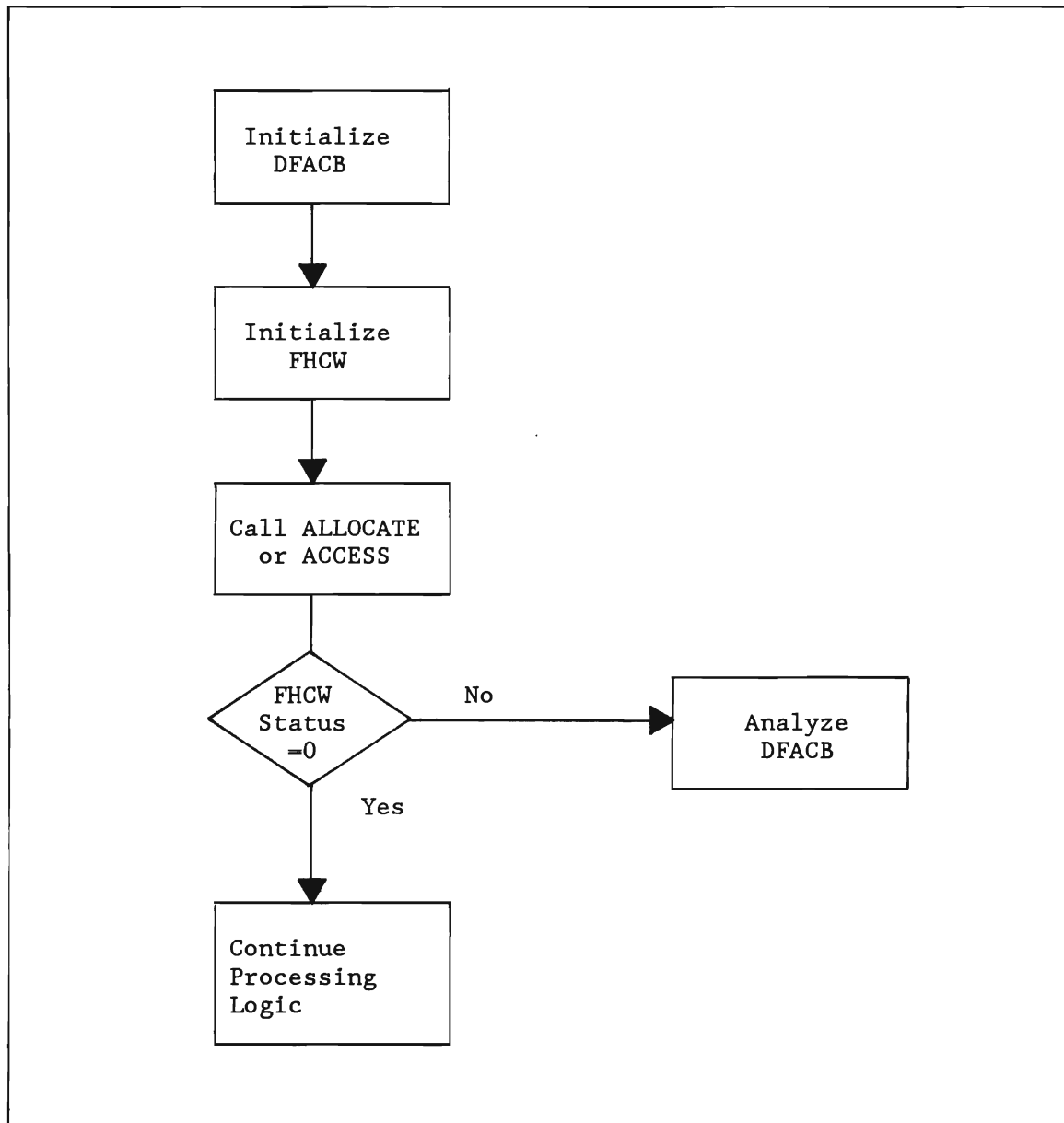


Figure 2. Using DFACB for Calls to ALLOCATE or ACCESS

All fields in the DFACB not being specifically supplied by the user should be set to binary zero. Assembler Language programs can use a Dsect of the DFACB via a COPY DDSASECT statement.

2.4 RETURN CODES

The return code from a call to ALLOCATE or ACCESS is given in the FHCW, as specified in Appendix A.4.

As shown in Figure 2, if the call completes with an error, the application subsystem may determine the error with greater accuracy by inspecting the additional diagnostic information in the DDSASTAT, DDSARETC and DDSASID fields.

2.5 DELETING DATA SETS

SAM data sets, whether dynamically accessed or allocated, may be optionally deleted when calling RELEASE by placing the character S in the second byte of the FHCW supplied to RELEASE. If the data set was accessed through the catalog, or if cataloging was requested when the data set was created using ALLOCATE, the data set will be uncataloged as well as scratched.

A call to ACCESS followed by a call to RELEASE (specifying the Scratch option) deletes a SAM data set even if no I/O has been performed.

VSAM data sets cannot be deleted or uncataloged by Dynamic File Allocation.

Chapter 3

JCL REQUIREMENTS AND INSTALLATION PROCEDURES

3.1 JCL REQUIREMENTS

If a SAM data set is not pre-existing and catalogued, each disk volume which will contain dynamically accessed or allocated data sets must be defined by a DD statement. The ddname must begin with the characters IXF. The DD statement should be coded in the following format:

```
//IXFxxxxx DD UNIT=unit,VOL=SER=volser,DISP=OLD[,SPACE=nnnn]
```

If a ddname is specified when calling ACCESS or ALLOCATE, the ddname must be present in the JCL. If a volume-serial number is supplied, the first DD statement which starts with IXF and which points to the indicated volume will be used. UNIT should specify the systemwide generic name for all direct access devices (SYSDA, etc). Omit the VOL parameter if a system work pack (STORAGE volume) is used.

A DYNAMNBR parameter on the Intercomm EXEC statement must be set to a number large enough for all concurrent allocations, including both system allocations and dynamic allocations. In most cases, the system default is large enough for the DYNAMNBR parameter to be omitted. Because a call to RELEASE for a dynamically accessed or allocated data set not only closes the DCB, but also deallocates the data set, coding of FREE=CLOSE on the DD statement is unnecessary.

3.2 INSTALLATION PROCEDURES

The procedure for installing Dynamic File Allocation is as follows:

1. Assemble and link IXFDYNAM, and add the module name to the Intercomm linkedit (see ICOMLINK macro, DFA parameter).
2. IXFDYNAM is Link Pack eligible and must be linked with IXFHND01 when the File Handler is Link Pack resident (see the Operating Reference Manual for installation details).
3. Relinkedit Intercomm, or the Link Pack Area, as applicable.



Appendix A

CALLS AND RETURN CODES

This appendix gives specifications of calls to DFA routines and their return codes. The following items are detailed:

- Calling syntax
- DFACB Specifications and return values
- Option codes
- Return codes

A.1 CALLING SYNTAX

<u>Assembler Language:</u> CALL {ALLOCATE},(extdsct,fhcn,dfacb),VL[,MF=(E,list)] {ACCESS }
<hr/> <u>Reentrant COBOL:</u> CALL 'COBREENT' USING routine-code, extdsct, fhcn, dfacb.
<hr/> <u>PL/1:</u> CALL PMIPL1 (routine-code,extdsct,fhcn,dfacb); or CALL {ALLOCATE} (extdsct,fhcn,dfacb); {ACCESS }

where:

routine-code is the name of the area containing the REENTSBS service routine code, as follows:

Service Routine	REENTSBS Code	ICOMSBS Name (COBOL)	PENTRY Name (PL/1)
ALLOCATE	75	DYN-ALLOCATE	ALLOCATE
ACCESS	71	DYN-ACCESS	ACCESS

extdsct is the standard 12-fullword File Handler External Data Set Control Table area to also be passed on subsequent calls to other File Handler routines.

fhcn is the fullword File Handler Control Word; a completion code is returned in the first byte indicating the success or failure of the subroutine call, as discussed below. For CALLS to ACCESS for VSAM files, a C'V' in byte 3 of the FHCW must be passed. To specify a VSAM file as READONLY, a C'R' in byte 2 of the FHCW must be passed.

dfacb is a 22-fullword Dynamic File Allocation Control Block area used to pass information between the calling program and IXFDYNAM. Appendix A.2 describes the fields within the DFACB and the information contained in these fields after the call is completed. (The field names used are the labels defined in Assembler Language subsystems.) The last two words of the DFACB are reserved. All fields not initialized by the user must be binary zeros.

A.2 DFACB Specifications and Return Values

Field/ Length (Bytes)/ Type	When Required	Specification
DDSADSN 44 character	Required for ACCESS and ALLOCATE.	Data set name (DSN). Left-justified and padded on right with blanks. Supplied DSN is unmodified upon return from call.
DDSADDNM 8 character	Required for ALLOCATE if DDSAVOLS is not supplied; DDSADDNM is preferred. Required for ACCESS if data set is not catalogued and DDSAVOLS is not supplied.	ddname of DD statement defining disk volume containing or to contain data set. Left-justified and padded on right with blanks. Upon return from call, field contains ddname of DD statement used to allocate or access data set. DDSADDNM or DDSAVOLS (but not both) must be supplied if the data set is not cataloged.
DDSAVOLS 6 character	Required for ALLOCATE if DDSADDNM is not supplied. Required for ACCESS if data set is not catalogued and DDSADDNM is not supplied.	Volume-serial of disk volume containing or to contain data set. Left-justified and padded on right with blanks. Upon return from call, field contains volume-serial of disk volume on which data set resides. DDSAVOLS or DDSADDNM (but not both) must be supplied if the data set is not cataloged.

(continued)

Field/ Length (Bytes)/ Type	When Required	Specification
DDSARECF* 2 character	Required for ALLOCATE.	For ALLOCATE, valid record formats (RECFM) for a data set are as follows: FF -- Fixed, unblocked FB -- Fixed, blocked VV -- Variable, unblocked VB -- Variable, blocked UU -- Undefined For ACCESS, set to binary zeros. Upon return from ALLOCATE, field is unmodified. Upon return from ACCESS, field contains two characters indicating record format, as above.
DDSALREC* 2 binary	Required for ALLOCATE.	For ALLOCATE, logical record length (LRECL) of data set. For ACCESS set to binary zeros. Upon return from ALLOCATE, field is unmodified. For ACCESS, LRECL of data set is returned.
DDSABLKS* 2 binary	Required for ALLOCATE.	For ALLOCATE, block size (BLKSIZE) of data set. For ACCESS, set to binary zeros. Upon return from ALLOCATE, field is unmodified. For ACCESS, BLKSIZE of data set is returned (except VSAM data sets).
DDSARSIZ 2 binary	Required for ALLOCATE if SPACE para- meter is not present on DD statement used to allocate data set.	Average record size of each physical block (normally equal to block size). Overrides supplied DD statement SPACE parameter. Upon return from ALLOCATE, field is unmodified. For ACCESS, set to binary zeros.
*Refer to IBM's <u>Data Administration Guide</u> for the interrelated values appropriate to these specific fields.		

(continued)

Field/ Length (Bytes)/ Type	When Required	Specification
DDSAPRIM 2 binary	Required for ALLOCATE, if SPACE parameter is not present on DD statement used to allocate data set.	For ALLOCATE, number of records to be accommodated in primary space allocation. Overrides supplied DD statement SPACE parameter. Upon return from ALLOCATE, field is unmodified. For ACCESS, set to binary zeros.
DDSASEC 2 binary	Not required even if SPACE parameter is not present on DD statement used to allocate data set.	Number of records to be accommodated in secondary space allocation. Overrides supplied DD statement SPACE parameter. Upon return from ALLOCATE, field is unmodified. For ACCESS, set to binary zeros.
DDSARETC 1 binary		Set to binary zero. Upon return from ALLOCATE or ACCESS, field contains return code of the last SVC issued by ALLOCATE or ACCESS routine.* It is useful in debugging and error determination.
DDSASID 1 character		Set to binary zero. Upon return from ALLOCATE or ACCESS, field contains a character identifying the last request issued as follows: <u>ALLOCATE</u> A--ALLOCATE (SVC 99) C--CATALOG R--RDJFCB <u>ACCESS</u> A--ALLOCATE L--LOCATE O--OBTAIN R--RDJFCB B--CLOSE ACB (VSAM) M--MODCB (VSAM) P--OPEN ACB (VSAM) S--SHOWCB (VSAM) T--TESTCB (VSAM)
*Refer to IBM documentation.		

(continued)

Field/ Length (Bytes)/ Type	When Required	Specification
DDSASTAT 4 character		<p><u>Byte 1</u>--Upon return from call, contains completion code. (See Appendix A.4.)</p> <p><u>Byte 2</u>--Option code. Upon return from call, this code remains unmodified. (See Appendix A.3.)</p> <p><u>Bytes 3-4</u>--For ALLOCATE, an option code can be set in byte 3 (See Appendix A.3.1.); for both ALLOCATE and ACCESS, on return from call, bytes 3 and 4 will contain either an information reason code or an error reason code, depending on successful or unsuccessful completion of the Allocate SVC.*</p>
DDSACINV 4 binary		VSAM only: for ACCESS, set to binary zeroes. Upon return from ACCESS, field contains CI size.
(filler) 8 binary		Set to binary zeros for both ACCESS and ALLOCATE.
<p>*Refer to IBM documentation; <u>OS/VS2 MVS SPL: Job Management</u>, or <u>MVS/XA SPL: System Macros and Facilities Volume 1</u>.</p>		

A.3 OPTION CODES

Option codes may be set in byte 2 (and byte 3, for ALLOCATE) of DDSASTAT, as described below.

A.3.1 ALLOCATE

Byte 2 of DDSASTAT.

Code	Function
(binary) 0	Data set is only to be allocated, and not cataloged.
(EBCDIC) C	Data set is to be cataloged as well as allocated. If this option is requested, all necessary catalog indices must already exist.

Byte 3 of DDSASTAT.

Code	Function
(binary) 0	Unused space is to be released when data set is closed.
(EBCDIC) N	Unused space is not be be released when data set is closed.

A.3.2 ACCESS

Byte 2 of DDSASTAT.

Code	Function
(binary) 0	The data set is accessed with a disposition of OLD and will be positioned at the beginning of the existing data. In this case, only calls to GET and READ are valid to retrieve the data and only calls to PUT (following a call to GET) are valid to update records.
(EBCDIC) M	The data set is accessed with an implied disposition of MOD. The data set will be positioned at the end of existing data so that subsequent calls to PUT or WRITE will extend the data set. If this option is requested, calls to GET or READ are invalid.
(EBCDIC) S	VSAM data set is accessed with a disposition of SHR.

A.4 RETURN CODES

On return from a call to ALLOCATE or ACCESS, the FHCW contains a return code as follows:

Code (EBCDIC Character)	Meaning
0	Call was successful.
9	Call was unsuccessful.

For a call to RELEASE, a return code of 1 in the FHCW indicates an invalid scratch requested for a VSAM file and was ignored; VSAM file successfully deallocated.

A more detailed return code is given in byte 1 of the DDSASTAT, as described below, and the fields DDSARETC and DDSASID as described in Appendix A.2.

A.4.1 ALLOCATE Return Codes in DDSASTAT

Code	Meaning
0	Normal completion
1	A ddname or volume-serial was supplied (in DDSADDNM or DDSAVOLS) but no DD statement of the proper form was found with the supplied ddname or which defined the supplied volume-serial. (See Chapter 3.)
3	Invalid or missing SPACE parameters (as supplied via DDSARSIZ, DDSAPRIM and DDSASEC).
4	Invalid or missing RECFM, LRECL or BLKSIZE values (as supplied in DDSARECF, DDSALREC and DDSABLKS).
5	No direct access space was available on the specified disk volume.
6	The data set could not be cataloged, but was allocated.
7	A data set with the specified data set name already exists on the indicated disk volume. The application may clear (to binary zeros) and reinitialize the DFACB for a call to ACCESS the existing data set.
8	An I/O error occurred during allocation.
9	Return Code 4 received from Allocate SVC 99.*
A	Return Code 8 received from Allocate SVC 99.*
B	Return Code 12 received from Allocate SVC 99.*
C	Return Code 16 received from Allocate SVC 99.*

* See also return codes in bytes 3 and 4 of DDSASTAT as described in Appendix A.2. For example, a return code of 9 in byte 1 of DDSASTAT accompanied by an error reason code of X'0448' in bytes 3 and 4 indicates the data set already exists; for corrective action, see return code 7.

A.4.2 ACCESS Return Codes in DDSASTAT

Code	Meaning
0	Normal completion
1	A ddname or volume-serial was supplied by the user (in DDSADDNM or DDSAVOLS) but no DD statement of the proper form was found with the supplied ddname or which defined the supplied volume-serial. (See Chapter 3.)
2	Neither DDSADDNM nor DDSAVOLS was supplied and the data set was either not cataloged or was cataloged with a volume-count greater than 1.
3	Neither DDSADDNM nor DDSAVOLS was supplied and the data set was cataloged on a volume not defined by a DD statement of the proper form. (See Chapter 3.)
4	The data set does not exist on the specified disk volume. The application program may clear (to binary zeros) and reinitialize the DFACB for a call to ALLOCATE a new data set.
5	The specified data set is not a sequential data set. Usually indicates an existing data set was incorrectly created, or the DFACB was not cleared (to binary zeros) and properly initialized before the call to ACCESS, or the DFACB area is not 22 fullwords (88 bytes) in length or not fullword aligned.
6	Return code 4 received from Allocate SVC 99.*
7	Return code 8 received from Allocate SVC 99.*
8	Return code 12 received from Allocate SVC 99.*
9	Return code 16 received from Allocate SVC 99.*
E	An error occurred in a VSAM macro. Can also occur if VSAM file specified but file is PS.
F	No core available for an ACB.

* See also return codes in bytes 3 and 4 of DDSASTAT as described in Appendix A.2.



Appendix B

DDSASECT DSECT

The following is the DDSASECT Dsect as released, giving Assembler Language specifications for the DFACB. In order to include this Dsect in an Assembler Language program, a COPY DDSASECT statement must be specified.

DDSASECT	DSECT		THIS DSECT MAPS THE DFACB SUPPLIED TO THE FILE
*			HANDLER BY APPLICATION PROGRAMS WHEN CALLING
*			ACCESS,ALLOCATE FOR DYNAMIC FILE ALLOCATION.
*			
*			
DDSADSN	DS	CL44	DATA SET NAME
DDSADDNM	DS	CL8	DDNAME
DDSAVOLS	DS	CL6	VOLUME-SERIAL
DDSARECF	DS	CL2	RECFM--FF,FB,VV,VB,UU
DDSALREC	DS	H	LOGICAL-RECORD-LENGTH
DDSABLKS	DS	H	DATA SET BLOCK SIZE
DDSASPAC	DS	OXL6	SPACE PARAMETERS
DDSARSIZ	DS	H	AVERAGE RECORD SIZE
DDSAPRIM	DS	H	# OF RECORDS IN PRIMARY ALLOCATION
DDSASEC	DS	H	# OF RECORDS IN SECONDARY ALLOCATION
DDSARETC	DS	X	RETURN CODE OF LAST SVC ISSUED
DDSASID	DS	X	ID OF LAST SVC ISSUED
DDSALOC	EQU	C'L'	LOCATE SVC
DDSAOBTN	EQU	C'O'	OBTAIN SVC
DDSAALOC	EQU	C'A'	ALLOCATE SVC
DDSACAT	EQU	C'C'	CATALOG SVC
DD SARJFC	EQU	C'R'	RDJFCB SVC
DDSASHOW	EQU	C'S'	SHOWCB (VSAM)
DDSATEST	EQU	C'T'	TESTCB (VSAM)
DDSAMODC	EQU	C'M'	MODCB (VSAM)
DDSAOPEN	EQU	C'P'	OPEN ACB (VSAM)
DDSACLOS	EQU	C'B'	CLOSE ACB (VSAM)
DDSASTAT	DS	CL4	ANCILLARY STATUS-AREA
*			
*			THE FOLLOWING RETURN-CODES ARE FOUND IN THE FIRST BYTE OF
*			DDSASTAT AFTER A CALL TO EITHER ACCESS OR ALLOCATE.
*			
DDSAOK	EQU	C'O'	NORMAL COMPLETION
DDSANODD	EQU	C'1'	DDNAME OR VOLSER SUPPLIED BUT
*			NO MATCH IN TIOT
*			

(continued)

*			THE FOLLOWING RETURN-CODES ARE FOUND IN THE FIRST BYTE OF
*			DDSASTAT AFTER A CALL TO ACCESS.
*			
DDSANGAT	EQU	C'2'	DATA SET EITHER NOT CATALOGED OR
*			VOLUME-COUNT GREATER THAN 1.
DDSABCAT	EQU	C'3'	DATA SET CATALOGED ON VOL-SER NOT
*			ACCESSABLE THROUGH TIOT.
DDSADSCB	EQU	C'4'	NO DSCB FOUND ON DISK FOR THE DATA SET.
DDSANTPS	EQU	C'5'	DATA SET TO BE ACCESSED IS NOT PHYSICAL
*			SEQUENTIAL (PS)
DDSABDVS	EQU	C'E'	ERROR IN VSAM MACRO
DDSANCOR	EQU	C'F'	NO CORE FOR ACB
DDSAAC04	EQU	C'6'	SVC 99 RC-04
DDSAAC08	EQU	C'7'	SVC 99 RC-08
DDSAAC0C	EQU	C'8'	SVC 99 RC-12
DDSAAC10	EQU	C'9'	SVC 99 RC-16
*			
*			THE FOLLOWING RETURN CODES ARE FOUND IN THE FIRST BYTE
*			OF DDSASTAT AFTER A CALL TO ALLOCATE.
*			
DDSANSPA	EQU	C'3'	INVALID OR MISSING SPACE PARAMETERS
DDSADCB	EQU	C'4'	INVALID OR MISSING RECFM,LRECL OR
*			BLOCKSIZE.
DDSAFULL	EQU	C'5'	NO DIRECT-ACCESS SPACE AVAILABLE
DDSAACAT	EQU	C'6'	DATA SET COULD NOT BE CATALOGED
DDSADUPL	EQU	C'7'	DATA SET ALREADY EXISTS
DDSAIOER	EQU	C'8'	I/O ERROR DURING ALLOCATION
DDSAAL04	EQU	C'9'	SVC 99 RC-04
DDSAAL08	EQU	C'A'	SVC 99 RC-08
DDSAAL0C	EQU	C'B'	SVC 99 RC-12
DDSAAL10	EQU	C'C'	SVC 99 RC-16
*			
*			THE FOLLOWING OPTION CODES ARE PASSED BY THE CALLER IN
*			THE SECOND BYTE OF DDSASTAT.
*			
DDSADCAT	EQU	C'C'	DATA SET SHOULD BE CATALOGED WHEN
*			ALLOCATED.
DDSAMOD	EQU	C'M'	DATA SET IS TO BE EXTENDED WHEN CALLING
*			ACCESS (EQUIVALENT TO DISP=MOD).
*			
DDSASHR	EQU	C'S'	DATA SET DISP - SHARE (VSAM FILES ONLY).
*			
*			THE FOLLOWING OPTION-CODE IS PASSED BY THE CALLER IN
*			THE THIRD BYTE OF DDSASTAT.
*			
DDSANRLS	EQU	C'N'	UNUSED SPACE IS NOT TO BE RELEASED
*			WHEN THE DATA SET IS CLOSED.
*			
DDSAACINV	DS	F	CONTROL INTERVAL SIZE (VSAM)
	DS	CL8	MUST BE ZERO
	DS	OD	
DDSALEN	EQU	*-DDSASECT	LENGTH OF THE DDSA AREA

INDEX

	<u>Page</u>		<u>Page</u>
ACCESS Service Routine		Error determination	5-6
--calls to	4, 10	Exclusive Control	1, 3
--and ddname	7	Execution JCL	1, 3, 4, 7
--deleting a data set	6	--and FREE=CLOSE	7
--description	1	Extending a data set	3
--and DFACB	5, 11-14	EXTDSCT. <u>See</u> External Data Set	
--extending a data set	4	Control Table.	
--file attributes	3	External Data Set Control Table	4, 10
--and File Handler service	2		
routines		FHCW. <u>See</u> File Handler Control Word.	
--option codes	15	FILE (system command)	1
--parameters	4	File Handler	3, 4, 7, 10
--and RELEASE	3, 6	File Handler Control Word	3, 4, 6, 10
--return codes	5-6, 14-17	File Handler Service	
		Routines	1-3, 15
ALLOCATE Service Routine		ICOMLINK--DFA parameter	7
--calls to	4, 10	Installation	7
--and ddname	7	IXFDYNAM module	7, 10
--description	1	JCL. <u>See</u> Execution JCL.	
--and DFACB	5, 11, 14	Link Pack installation	7
--and File Handler Service		Option codes	4, 14-15
Routines	2	Parameters	4
--option codes	14-15	--for calls	10
--parameters	4	REENTSBS Codes	10
--and RELEASE	3, 6	RELEASE Service Routine	1, 3
--return codes	5, 6, 14-17	--and scratch option	6
		Return codes	6, 14-17
Allocation. <u>See</u> Space Allocation.		Scratching a data set	6
Application design		Space Allocation	3,7
--considerations for	3	--DFACB fields for	12-13
Cataloging dynamically allocated		--secondary	3
data sets	1, 4, 6, 15	--and x37 abends	3
Creating dynamically allocated		Updating a dynamically allocated	3
data sets	1, 4	data set	
DDSASECT Dsect	5, 19-20	Volume-serial specification not	3
Deleting dynamically allocated		recommended	
data sets	6	VSAM data sets	1-4,6
DFACB. <u>See</u> Dynamic File Allocation		--FHCW codes	10
Control Block.		--return codes	13, 15, 17, 19-20
Disposition of dynamically		--Control Interval (CI) size	3,14,20
allocated data sets	1-4, 15	x37 abends	3
Dynamic File Allocation			
Control Block	4-6, 10, 11		
Dynamic File Allocation Service			
Routines			
--calls to	4, 10		
-- <u>See also</u> ACCESS and ALLOCATE.			
DYNAMNBR parameter	7		
Enqueue/Dequeue Facility	3		

