



DOCUMENT INTERCHANGE ARCHITECTURE:  
APPLICATION PROCESSING SERVICES REFERENCE

SC23-0761-0



**DOCUMENT INTERCHANGE ARCHITECTURE:  
APPLICATION PROCESSING SERVICES REFERENCE**

### **First Edition (June 1983)**

Changes are made periodically to the information herein; any such changes will be reported in subsequent revisions or Technical Newsletters.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Department D31, 11400 Burnet Road, Austin, Texas 78758. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

## PREFACE

IBM is committed to providing document interchange capabilities across a broad spectrum of IBM products designed for use in office systems. The strategic IBM architecture for accomplishing this interchange is the Document Interchange Architecture.

The Document Interchange Architecture (DIA) is a program-to-program communication architecture. Specifically, DIA defines the protocols and data structures that enable programs to communicate processing intentions and to interchange data. DIA logically divides into several parts: an information interchange base and various DIA application services.

This manual describes the basic concepts, protocols, and data structures to perform the DIA application processing service. The Application Processing Services encompasses the protocols and commands to remotely perform jobs and tasks including facilities to request functions such as execute, format, and modify.

This manual is intended for data processing managers, system analysts, designers, system programmers, and application programmers, as well as systems engineers and product support representatives.

## PREREQUISITE PUBLICATION

Document Interchange Architecture: Concepts and Structures, SC23-0759.

## RELATED PUBLICATIONS

- Office Information Architecture: Concepts, GC23-0765-0
- Document Interchange Architecture: Document Distribution Services Reference, SC23-0762
- Document Interchange Architecture: Document Library Services Reference, SC23-0760
- Document Interchange Architecture: Interchange Document Profile Reference, SC23-0764
- Document Interchange Architecture: Transaction Programmer's Guide, SC23-0763
- Document Content Architecture: Revisable-Form-Text Reference, SC23-0758
- Document Content Architecture: Final-Form-Text Reference, SC23-0757
- Systems Network Architecture: Concepts and Products, GC30-3072.



# CONTENTS

## Chapter 1. Introduction 1

## Chapter 2. Functional Description 3

Logical Components of an Office System Network 3

Application Processing Services Concepts 4

Document Interchange Unit 4

Commands and Control Flow 5

Command Classes 5

Request-Reply Protocols 6

Function Sets 7

Function Set Negotiation 7

Function Set Definition 7

## Chapter 3. Command Descriptions 9

DELIVER 9

EXECUTE 13

FORMAT 16

MODIFY 21

## Appendix A. Operand Descriptions 27

CORRELATION (Format 1) 27

FORMATTED-DOCUMENT-NAME (Format 1) 28

FORMATTER-NAME (Format 1) 28

FORMAT-PARAMETERS (Format 1) 28

IDENTIFIED-DATA (Format 1) 29

IDENTIFIED-DATA (Format 3) 29

IDENTIFIED-DATA (Format 42) 30

MODIFY-DATA (Format 41) 31

ORIGINATING-NODE-ADDRESS (Format 1) 33

PROCESS-PARAMETERS (Format 1) 33

PROCESS-NAME (Format 1) 33

PROCESS-PASSWORD (Format 1) 34

SOURCE-ADDRESS (Format 1) 34

SOURCE-ADDRESS (Format 42) 34

SOURCE-PASSWORD (Format 1) 35

## Appendix B. Document Interchange Unit (DIU) 37

DIU Structured Field 37

DIU Prefix 39

Command Sequence 40

Data Units 41

Document Units 42

DIU Suffix 43

## Appendix C. DIA Code Points 45

## Appendix D. DIA Document Types 51

## Glossary 53



## FIGURES

1. Function Set 9 8
2. DIU Structured Field 38
3. DIU Introducer (INTRO) 38
4. Document Type Code Assignments 51



## CHAPTER 1. INTRODUCTION

Document Interchange Architecture defines the protocols and data streams necessary to interchange information such as documents and messages in a consistent, predictable manner.

DIA provides these categories of services for the interconnection of office systems:

- Document Library Services
- Document Distribution Services
- Application Processing Services.

**Document Library Services** allow users to file documents in a document library, to retrieve them or delete them from the library, and to search the library for documents that meet user-specified criteria, such as the the name of the author. These criteria are compared with document descriptors that are stored with the document. The user can obtain all documents filed in the document library that meet those criteria.

**Document Distribution Services** deliver documents and related information from their source to one or more recipients anywhere in the network. These services can, for example, allow a user to enter a single request to distribute a document to multiple recipients, schedule distribution by document priority, confirm delivery, and report errors. Document Distribution Services are commonly referred to as electronic document distribution.

**Application Processing Services** allow users to modify document descriptors used in searching a library; to invoke a program to transform documents from one format to another, for example, revisable-form-text to final-form-text; and to execute user-supplied programs. This reference manual contains detailed information about the Application Processing Services of DIA.

Chapter 2 begins with a review of DIA concepts applicable to Application Processing Services and concludes with the function set which defines the Application Processing Services.

Detailed specifications for the commands that comprise the Application Processing Services are presented in Chapter 3. Any exception conditions that are unique to each command are listed with the command description.

The reference manual concludes with an appendix which contains details about each operand used in Application Processing Services; an appendix which describes the Document Interchange Unit, which is the major unit of interchange in DIA; and an appendix which gives a table of encodings for the DIA entities which are applicable to Application Processing Services.



## CHAPTER 2. FUNCTIONAL DESCRIPTION

Document Interchange Architecture defines a set of services. These services are performed by processes implemented in the uppermost layer of a communication architecture such as IBM's Systems Network Architecture (SNA). DIA specifies how these processes, located throughout a network, communicate with each other to perform required office system functions.

Each DIA service performs specified functions requested by end users. An end user represents the source or the recipient of information flowing through the office system network. Each end user of a DIA process is uniquely identified in the network by a logical address.

The information exchanged by DIA services comprises DIA commands and user information. Typical commands are: distribute a document from office system A to office systems B, C, and D; retrieve document XYZ from the document library; and search the document library for documents that satisfy search criteria J, K, and L.

Document Interchange Architecture is considered a part of SNA. (Only that part of SNA is introduced by this book; SNA as a whole is described in SNA Concepts and Products.) However, DIA is not dependent on the specific presentation and transport services of the network, and is not concerned with the content of the documents being interchanged among office systems.

### LOGICAL COMPONENTS OF AN OFFICE SYSTEM NETWORK

A network of office systems based on Document Interchange Architecture contains a set of interrelated logical components which lie within the physical components of the network. The logical components are defined by the DIA and are implemented by IBM products as processes executing in physical components. These logical components are:

- A source node which acts on behalf of an end user to provide DIA services that initiate and control the interchange of documents and other information with end users called recipients.
- A recipient node which acts on behalf of an end user (recipient) to provide DIA services that control and receive documents and other information sent by a source node or an office system node.
- An office system node (OSN) which provides DIA services that receive, store, route, and deliver information for source and recipient nodes. An OSN contains storage capabilities providing the document library for attached source nodes. An office system node can also interact with an appropriately configured network to distribute information to other office system nodes.

Source nodes, recipient nodes, and office system nodes interchange documents and other information through an office system network using the transport services of the network. The nodes are uniquely identified in the network.

Specifically, a source node is identified by a source address, a recipient node is identified by a recipient address, and an office system node is identified by either an originating node address or a destination node address. An OSN is an originating node when it supports a source node and is a destination node when it supports a recipient node.

Originating node addresses and destination node addresses are unique within the network. Source and recipient addresses are unique within originating nodes and destination nodes, respectively.

An OSN process can act both as an originating node and a destination node concurrently. In this case, the originating node address and the destination node address are identical. Similarly, a DIA process can act in the capacity of both a source node and a recipient node. In this mode, the values of the source address and recipient address are identical.

## APPLICATION PROCESSING SERVICES CONCEPTS

Application processing services define commands that cause an office system node to perform several additional functions. These additional functions allow end users to manipulate document descriptors associated with a document (for example, to add or delete the descriptors); to invoke a program to transform documents from revisable-form-text to final-form-text; and to invoke specific application programs, procedures, or processes.

An application processing services source node provides the following functions for end users:

- Requests execution of programs within the office system node
- Requests the modification of descriptors defined in a document profile
- Invokes programs to format documents.

An application processing services office system node provides functions requested by end users at source nodes. These functions are:

- Interprets and validates requests from the source nodes
- Modifies document descriptors specified by end users
- Schedules execution of programs and procedures requested by end users
- Executes programs to transform documents from revisable-form-text to final-form-text.

## DOCUMENT INTERCHANGE UNIT

The basic unit of interchange exchanged between DIA processes is the document interchange unit (DIU). A DIU is made up of the following data stream components:

PREFIX	COMMAND SEQUENCE	DATA UNITS	DOCUMENT UNITS	SUFFIX
--------	---------------------	---------------	-------------------	--------

- The prefix contains the information to introduce and identify the DIU.
- The command sequence contains the command that specifies the function to be performed and related processing information.
- The data unit contains information that may be referred to by the DIA command in the command sequence. This field is optional and is present when defined by the command.
- The document unit contains the document profile and may contain the document content. This field is optional and is present only when a document profile and content are sent from one DIA process to another.
- A suffix specifies the end of the DIU and indicates whether any abnormal conditions occurred while the DIU was being transmitted.

These data stream components may be composed of substructures called subcomponents. Examples of subcomponents are command operands and document profiles. All DIU components and their subcomponents begin with a structured field called an introducer. The introducer uniquely identifies each field and indicates its length. Consequently, all fields and components (and hence the entire data stream) are self-describing and may be variable in length.

## COMMANDS AND CONTROL FLOW

Control flow between DIA processes is command driven. Commands are divided into classes, depending upon the protocol of the expected reply. The command class is identified by the encoding used in the class byte of the command. Commands that are used to reply to a previous command include a CORRELATION operand to correlate the reply to the previous command. Command classes and the correlation of replies to commands are described in the following two sections.

### Command Classes

This section contains detailed information describing each command class defined for the architecture. The three classes of commands are as follows:

- No Reply Required Command Class (NRR)

The NRR command class is used for any command that does not require a replying command from the receiver. No explicit synchronization or correlation is done for this type of command exchange. Only an ACKNOWLEDGE

with an exception condition code is allowed to reply to and reference this class of command. The exception condition information can be used for statistics logging.

- Synchronous Reply Required Command Class (SRR)

The SRR command class is used for any command that requires a replying command as the next command sent by the receiver. For normal nonpreemptive command flow, the SRR requesting command sender should not send any other commands until the replying command has been received. The replying command may be any command that has a CORRELATION operand correlating it with the SRR command.

- Asynchronous Reply Required Command Class (ARR)

The ARR command class is used for any command that requires a replying command within the following conditions:

- The reply need not be the next command sent by the receiver.
- The reply need not be sent during the current DIA session.
- Replies to ARR commands may be sent in any order.
- Replies to ARR commands may not be received in the order sent by the ARR request processor.

The ARR commands must be replied to with a command that has a CORRELATION operand correlating it with the ARR command.

Normal termination of the DIA session while a reply to an ARR command is outstanding does not affect either the ARR command or the reply to that command.

An individual command may be used in any command class except where specifically excluded by its command definition.

When an exception condition is encountered in the processing of a command in any class, an ACKNOWLEDGE command with an exception condition code will be returned in the replying command class to the sender of the command exception.

Any command with a CORRELATION operand may be used as a replying command. The CORRELATION operand will contain the information which identifies the command to which this replying command is being sent. The correlation information is constructed from the DIU identifier found in the prefix. This information is used by the command requestor to confirm and manage the conclusion and disposition of requested process results.

## Request-Reply Protocols

The flow of DIUs between two DIA processes is command-driven and complies with the command or reply protocols specified for each command defined by DIA. The following general rules apply to the issuing of commands within the DIA Session:

- Only one partner in a particular DIA session can be sending command requests at any one time.
- The DIA partner which sends the SIGN-ON request is the partner which sends the first command in the DIA session.

The mapping of DIA flow control protocols to the use of verbs at the SNA LU 6.2 protocol boundary is described in Transaction Programmer's Guide. The mapping of DIA flow control to other communication subsystem implementations is described in the publications of the implementing products.

## FUNCTION SETS

Because office systems vary in their capabilities, DIA commands are grouped into function sets that identify the scope of work for a DIA session. These function sets have been defined so that each set contains all the commands required for a well-defined, usable, and complete set of functions for a given category of services.

### Function Set Negotiation

DIA processes establish a logical connection, called a DIA session, through which they exchange information. The DIA session exists after the two DIA processes identify themselves and agree on the scope of work that is to be performed. This agreement is necessary because not all DIA implementations support the same range of functions. DIA defines a wide range of office system functions; most office systems require only a subset of these functions for their operation.

The negotiation includes the determination of the roles each process will play. The process that will be the command requestor is identified as Process B and the process which will be the command server is identified as Process A. In the case of symmetric interchange, for example, when DIA processes are capable of simultaneously acting as a requestor and a server of a DIA function, the DIA process must assume the role of both Processes A and B.

### Function Set Definition

A brief summary of the Application Processing Services commands follows:

- The DELIVER command transports a document from a server node to a requestor node.
- The EXECUTE command requests an office system node to invoke the named process for execution.
- The FORMAT command requests an office system node to execute the named formatting process using the identified document for the format input object.

- The MODIFY command requests an office system node to revise document control information fields. A field must be uniquely identified as a specific occurrence of a DIA parameter with an assigned code point and format.

Figure 1 is a table showing how these commands are grouped into the Application Processing Services function set. The figure lists each command for the function set and identifies the valid command class for each command and the request/reply protocol. The request/reply protocol is represented by send or receive in the columns for Process A and Process B. Send indicates that the process is the command requestor and receive indicates the command server. Support of any function set requires that a DIA process assuming the role of either Process A or Process B must recognize and process all the commands designated as receive.

COMMAND	COMMAND CLASS	PROCESS A (SERVER)	PROCESS B (REQUESTOR)
EXECUTE	SRR	receive	send
FORMAT	SRR	receive	send
MODIFY	SRR	receive	send
DELIVER	NRR	send	receive
ACKNOWLEDGE	NRR	send/rec	send/rec
SIGN-ON Request	SRR	receive	send
SIGN-ON Reply	NRR	send	receive
SIGN-OFF	NRR	send/rec	send/rec

Figure 1. Function Set 9

## CHAPTER 3. COMMAND DESCRIPTIONS

This section contains the Application Processing Services command descriptions. Each command description begins with the command name and a list of the command operands. Optional operands are denoted by brackets. Required operands are shown without brackets.

The function of the command is explained, followed by a description of each operand. The detailed operand definitions are contained in "Appendix A. Operand Descriptions" on page 27. The command structured field identifiers (IDF) are defined in "Appendix C. DIA Code Points" on page 45.

Each command description contains the request/reply protocol used between the command requestor and command server. Normal and exception condition scenarios are shown.

The command descriptions are concluded with a list of exception conditions which are specific to the command. The general exceptions that are common to all DIA commands are described in "DIU General Exception Conditions" in Concepts and Structures.

The command descriptions for the ACKNOWLEDGE, SIGN-ON, and SIGN-OFF commands are contained in Concepts and Structures.

### DELIVER

<u>Command</u>	<u>Operands</u>
DELIVER	IDENTIFIED-DATA, CORRELATION

The DELIVER command is used as a replying command to return a document from a command server to the command requestor. An example of this is returning a formatted document in reply to a FORMAT command.

## Operand Descriptions

### IDENTIFIED-DATA

The IDENTIFIED-DATA (Format 1) operand specifies the location of the data to be delivered, specifically, the DIU Document Unit.

### CORRELATION

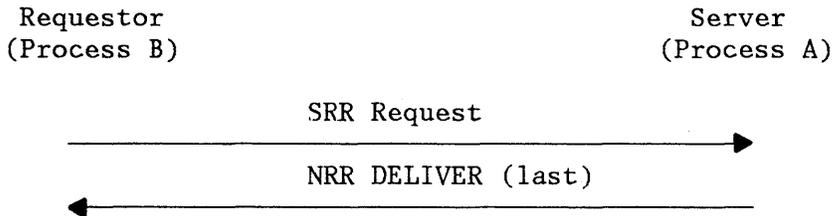
The CORRELATION (Format 1) operand is used to correlate a replying command to a previously sent request. The CORRELATION (Format 1) operand identifies the request to which the command is replying and gives an indication of whether or not additional replying commands are to be expected, that is, a last or not-last indicator is returned. When the last replying command has been received, the request is considered complete.

## Request/Reply Protocol

The following scenarios illustrate possible replies using the DELIVER command.

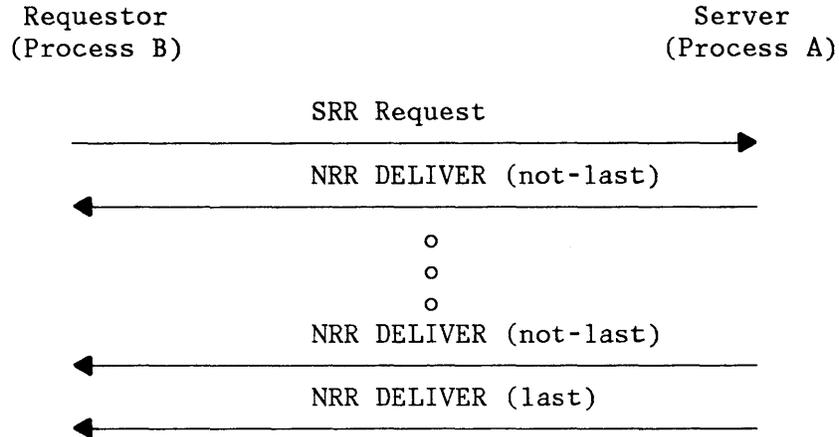
- Scenario 1 - Single Reply

The following is a single request and reply scenario.



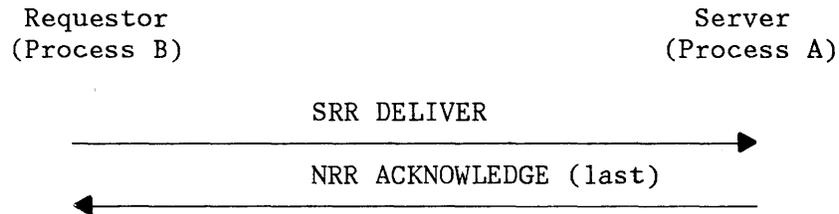
- Scenario 2 - Multiple Replying Commands

The following is a multiple replying command scenario.



- Scenario 3 - Exception Conditions.

Exception conditions detected during the DELIVER command processing will be replied to with an ACKNOWLEDGE command that contains the exception condition in the EXCEPTION-CODE operand.



### Exception Conditions

The general exception conditions that are common to all DIA commands are described in "DIU General Exception Conditions" in Concepts and Structures. The DELIVER command will be rejected by the receiver if any of the following conditions exist.

- The receiver is not in a state in which it can receive data.

Exception = Catastrophic, Session, Intervention-Required, Unknown  
 Exception Code = X'C11217'

- The IDENTIFIED-DATA operand has been omitted.

Exception = Catastrophic, Syntax, Data-Not-Found, Command-Operand  
 Exception Code = X'C20708'

- The CORRELATION operand does not reference a command previously sent by the receiver.

Exception = Catastrophic, Semantic, Data-Not-Found, Operand-Value  
Exception Code = X'C30709'  
Exception Data = LLIDF and data of CORRELATION operand

- The IDENTIFIED-DATA operand references non-existent data.

Exception = Catastrophic, Semantic, Data-Not-Found, Operand-Value  
Exception Code = X'C30709'  
Exception Data = LLIDF and data of IDENTIFIED-DATA operand

- The Document Unit Type is not supported.

Exception = Catastrophic, Semantic, Data-Not-Supported, Document-Unit  
Exception Code = X'C3020C'  
Exception Data = LLIDF of Document Unit Introducer

- The Document Content Introducer type is not supported.

Exception = Catastrophic, Semantic, Data-Not-Supported,  
Document-Content-Introducer  
Exception Code = X'C30210'  
Exception Data = LLIDF of Document Content Introducer

- The Document Type in the Document Unit ID is not supported.

Exception = Catastrophic, Semantic, Data-Not-Supported, Document-Unit ID  
Exception Code = X'C3020D'  
Exception Data = LLIDF and data of Document Unit ID

- The Document Type parameter in the Base Subprofile is not supported.

Exception = Catastrophic, Semantic, Data-Not-Supported,  
Document-Profile-Parameter  
Exception Code = X'C3020F'  
Exception Data = LLIDF and data of Document Type parameter

- The document profile is not supported.

Exception = Catastrophic, Semantic, Data-Not-Supported, Document-Profile  
Exception Code = X'C3020E'  
Exception Data = LLIDF of Document Profile

- The receiving processes resources are unavailable.

Exception = Catastrophic, Process, Resource-Not-Available, Document-Unit  
Exception Code = X'C4040C'

- The receiving process cancels the delivery of the data.

Exception = Catastrophic, Process, Cancelled, Command  
Exception Code = X'C41407'

## EXECUTE

<u>Command</u>	<u>Operands</u>
EXECUTE	PROCESS-NAME [, IDENTIFIED-DATA] [, ORIGINATING-NODE-ADDRESS] [, SOURCE-ADDRESS] [, SOURCE-PASSWORD] [, PROCESS-PARAMETERS] [, PROCESS-PASSWORD]

The EXECUTE command schedules a named program or procedure for asynchronous execution.

Input data may optionally be passed to the scheduled program or procedure for processing. The input data to be processed may either be submitted with the processing request (DIU) or it may be located in the server's document library. Further, input processing parameters may also be passed to the program or procedure when the program or procedure is invoked.

Output from the asynchronously scheduled program or procedure may be returned to the requestor using the DIA Document Distribution Services or non-DIA facilities.

### Operand Descriptions

#### PROCESS-NAME

The PROCESS-NAME (Format 1) operand specifies the 1- to 32-character name of the program or procedure to be invoked.

#### IDENTIFIED-DATA

The IDENTIFIED-DATA operand, if present, specifies the location of the data to be processed by the scheduled program or procedure. The data to be processed may be supplied with the request (DIU) using IDENTIFIED-DATA (Format 1) or referenced in the server's document library using IDENTIFIED-DATA (Format 3 or 42). If omitted, no input data will be passed for processing.

## ORIGINATING-NODE-ADDRESS

The ORIGINATING-NODE-ADDRESS (Format 1) operand, if present, specifies the 1- to 8-character group address token of the requestor which initiated this request. If omitted, the group address token of the command sender is assumed.

## SOURCE-ADDRESS

The SOURCE-ADDRESS operand, if present, specifies the element address token of the requestor which initiated this request; either SOURCE-ADDRESS (FORMAT 1 or 42) is valid. If omitted, the element address token of the command sender is assumed.

## SOURCE-PASSWORD

The SOURCE-PASSWORD (Format 1) operand specifies the 1- to 8-character access authorization key of the requestor which initiated this request. The SOURCE-PASSWORD (Format 1) operand is conditionally required if the SOURCE-ADDRESS (Format 1) is specified. The SOURCE-PASSWORD (Format 1) operand is required if:

- The SOURCE-ADDRESS (Format 1) operand specifies an element address token different than the command sender.
- The specified requestor does not have affinity with the command sender.
- The specified requestor has a password.

Otherwise, the SOURCE-PASSWORD should be omitted.

## PROCESS-PARAMETERS

The PROCESS-PARAMETERS (Format 1) operand, if present, specifies 1- to 32,767-bytes of input parameters that may be passed to the named program or procedure to control execution options. The syntax of the input parameters are defined by the named process or procedure. If omitted, no input parameters are passed when the program or procedure is invoked.

## PROCESS-PASSWORD

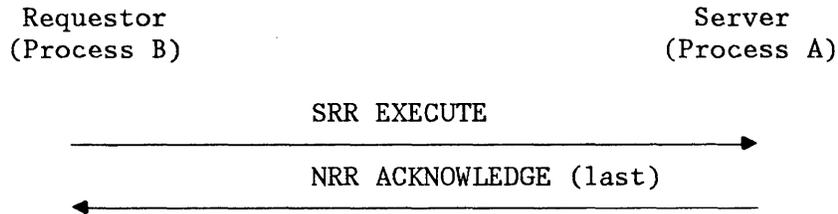
The PROCESS-PASSWORD (Format 1) operand, if present, specifies the 1- to 8-character authorization key that permits executing the named process.

## Request/Reply Protocol

The following scenarios illustrate possible replies to the EXECUTE command:

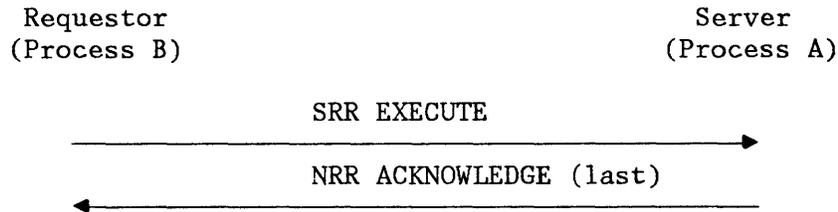
- Scenario 1 - Normal Completion

The reply command to an EXECUTE command is an ACKNOWLEDGE command that is sent to the requestor when the named program or procedure is successfully scheduled for execution.



- Scenario 2 - Exception Conditions

Exception conditions detected during the EXECUTE command processing will be replied to with an ACKNOWLEDGE command that contains the exception condition in the EXCEPTION-CODE operand.



### Exception Conditions

The general exceptions that are common to all DIA commands are described in "DIU General Exception Conditions" in Concepts and Structures. The following exception conditions are specific to the EXECUTE command and are detected and reported in addition to the general exception conditions.

- The SOURCE-PASSWORD operand is present and the SOURCE-ADDRESS (Format 1) operand is not present.

Exception = Catastrophic, Syntax, Data-Not-Found, Command-Operand  
 Exception Code = X'C20708'  
 Exception data = LLIDF of SOURCE-ADDRESS (Format 1) operand

- The SOURCE-ADDRESS operand identifies a requestor different than the command sender; the specified requestor does not have affinity with the command sender; the specified requestor has a password; and the SOURCE-PASSWORD operand is not specified.

Exception = Catastrophic, Semantic, Data-Not-Found, Command-Operand  
 Exception Code = X'C30708'  
 Exception data = LLIDF of SOURCE-PASSWORD operand

- The SOURCE-ADDRESS operand contains an invalid address.  
 Exception = Catastrophic, Semantic, Data-Not-Supported, Operand-Value  
 Exception Code = X'C30209'  
 Exception data = LLIDF of SOURCE-ADDRESS operand and data
- The ORIGINATING-NODE-ADDRESS operand contains an invalid address.  
 Exception = Catastrophic, Semantic, Data-Not-Supported, Operand-Value  
 Exception Code = X'C30209'  
 Exception data = LLIDF of ORIGINATING-NODE-ADDRESS operand and data
- The SOURCE-PASSWORD operand contains an invalid authorization key.  
 Exception = Catastrophic, Semantic, Password-Invalid, Operand-Value  
 Exception Code = X'C30509'  
 Exception data = LLIDF of SOURCE-PASSWORD operand and data
- The process or procedure name specified in the PROCESS-NAME operand cannot be found.  
 Exception = Catastrophic, Process, Data-Not-Found, Operand-Value  
 Exception Code = X'C40709'  
 Exception data = LLIDF of PROCESS-NAME operand and data

## FORMAT

<u>Command</u>	<u>Operands</u>
FORMAT	IDENTIFIED-DATA, FORMATTER-NAME [,FORMATTED-DOCUMENT-NAME] [,ORIGINATING-NODE-ADDRESS] [,SOURCE-ADDRESS] [,SOURCE-PASSWORD] [,FORMAT-PARAMETERS]

The FORMAT command is used to invoke a formatter to transform data from one format to another format, for example, a revisable-form-text document into a final-form-text document.

The data to be formatted may either be submitted with the request (DIU) or may be located in the server's document library.

The output from the formatting program may be either returned to the requestor or filed in the server's document library.

### Operand Descriptions

#### IDENTIFIED-DATA

The IDENTIFIED-DATA operand specifies the location of the data to be formatted. The data to be formatted may be supplied with the request using IDENTIFIED-DATA (Format 1) or referenced in the server's document library using IDENTIFIED-DATA (Format 3 or 42).

#### FORMATTER-NAME

The FORMATTER-NAME (Format 1) operand specifies the 1- to 44-character name of the formatting program to be invoked.

#### FORMATTED-DOCUMENT-NAME

The FORMATTED-DOCUMENT-NAME (Format 1) operand, if present, specifies the 1- to 44-character name to be assigned to the formatted data that is stored in the server's document library. If the operand is omitted, the formatted data is returned to the requestor using the DELIVER command.

#### ORIGINATING-NODE-ADDRESS

The ORIGINATING-NODE-ADDRESS (Format 1) operand, if present, specifies the 1- to 8-character group address token of the requestor which initiated this request. If omitted, the group address token of the command sender is assumed.

#### SOURCE-ADDRESS

The SOURCE-ADDRESS operand, if present, specifies the element address token of the requestor which initiated this request; either SOURCE-ADDRESS (FORMAT 1 or 42) is valid. If omitted, the element address token of the command sender is assumed.

#### SOURCE-PASSWORD

The SOURCE-PASSWORD (Format 1) operand specifies the 1- to 8-character access authorization key of the requestor which initiated this request. The SOURCE-PASSWORD (Format 1) operand is conditionally required if the SOURCE-ADDRESS (Format 1) is specified. The SOURCE-PASSWORD (Format 1) operand is required if:

- The SOURCE-ADDRESS (Format 1) operand specifies an element address token different than the command sender.
- The specified requestor does not have affinity with the command sender.
- The specified requestor has a password.

Otherwise, the SOURCE-PASSWORD should be omitted.

#### FORMAT-PARAMETERS

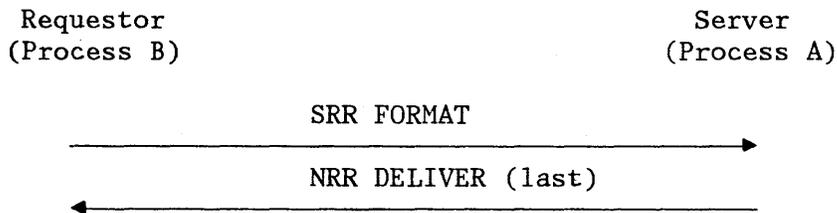
The FORMAT-PARAMETERS (Format 1) operand, if present, specifies 1- to 32,762-bytes of input data parameters that are passed to the named formatting program. The input data parameters are used to control the format processing options within the formatting program. The syntax of the input data is defined by the formatting program. If omitted, the formatting program is invoked without input data parameters.

#### Request/Reply Protocol

The following scenarios illustrate possible replies to the FORMAT command:

- Scenario 1 - Formatted data returned to the requestor

If the FORMATTED-DOCUMENT-NAME operand is omitted, the output from the formatting program is returned to the requestor using one or more replying DELIVER command(s). The number of replying DELIVER commands is determined by the formatting program that is invoked.



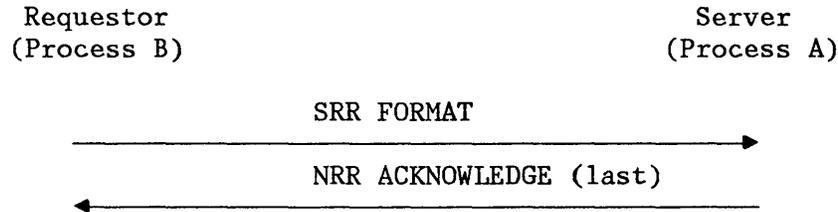
Since the FORMAT command is a specific request, the document(s) returned by the formatting program are not subject to the filtering or transformation defined when the command sender's DIA session was established.

- Scenario 2 - Formatted data filed in the document library

If the FORMATTED-DOCUMENT-NAME operand is specified, the formatted data will be filed in the server's document library according to the semantics of the DIA Document Library Services FILE command. The filed document will be assigned a library assigned document name (LADN), stored as a private document, and the requestor of the FORMAT command defined as the primary owner of the document. The name supplied in the FORMATTED-DOCUMENT-NAME operand is used as the base subprofile Document Name parameter when the document is filed into the document library. The formatting program is also responsible for specifying the Document-Type and Profile GCID base subprofile parameters prior to filing the document into the document library.

The replying command to a filed formatted document is an ACKNOWLEDGE command with no exception code indications in the EXCEPTION-CODE operand and the

LADN (Library Assigned Document Name) returned in the REPLY-DATA operand. The file server generates the LADN for the formatted document by concatenating the document library node address token and the date and time that the file process was successfully completed.

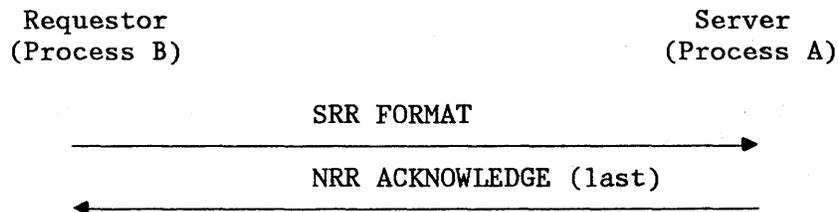


The format of the REPLY-DATA operand is defined as follows:

REPLY-DATA	LIBRARY ASSIGNED DOCUMENT NAME
INTRODUCER	INTRODUCER
LLIDF	LLIDF - IDD (Format 42)
X'nnnnC34501'	X'nnnnC32042' (IDD (Format 42) Introducer)
	LT X'0A01' Date and Time
	X'YYMDhms'hs'
	LT X'nn02' Document Library Node address
	C'ccc . . . c' 1- to 8-byte character string

- Scenario 3 - Exception Conditions.

Exception conditions detected during the FORMAT command processing will be replied to with an ACKNOWLEDGE command that contains the exception condition in the EXCEPTION-CODE operand.



## Exception Conditions

The general exception conditions that are common to all DIA commands are described in "DIU General Exception Conditions" in Concepts and Structures. The following exception conditions are specific to the FORMAT command and are detected and reported in addition to the general exception conditions.

- The SOURCE-PASSWORD operand is present and the SOURCE-ADDRESS (Format 1) operand is not present.

Exception = Catastrophic, Syntax, Data-Not-Found, Command-Operand  
Exception Code = X'C20708'  
Exception data = LLIDF of SOURCE-ADDRESS (Format 1) operand

- The SOURCE-ADDRESS operand specifies a requestor different than the command sender; the specified requestor does not have affinity with the command sender; the specified requestor has a password; and the SOURCE-PASSWORD operand is not specified.

Exception = Catastrophic, Semantic, Data-Not-Found, Command-Operand  
Exception Code = X'C30708'  
Exception data = LLIDF of SOURCE-PASSWORD operand

- The SOURCE-ADDRESS operand contains an invalid address.

Exception = Catastrophic, Semantic, Data-Not-Supported, Operand-Value  
Exception Code = X'C30209'  
Exception data = LLIDF of SOURCE-ADDRESS operand and data

- The ORIGINATING-NODE-ADDRESS operand contains an invalid address.

Exception = Catastrophic, Semantic, Data-Not-Supported, Operand-Value  
Exception Code = X'C30209'  
Exception data = LLIDF of ORIGINATING-NODE-ADDRESS operand and data

- The SOURCE-PASSWORD operand contains an invalid authorization key.

Exception = Catastrophic, Semantic, Password-Invalid, Operand-Value  
Exception Code = X'C30509'  
Exception data = LLIDF of SOURCE-PASSWORD operand and data

- The document specified in the IDENTIFIED-DATA operand cannot be found.

Exception = Catastrophic, Process, Data-Not-Found, Operand-Value  
Exception Code = X'C40709'  
Exception data = LLIDF of IDENTIFIED-DATA operand and data

- The specified requestor is not an owner or owner-delegate of the document specified in the IDENTIFIED-DATA operand. The requested service cannot be executed without the requestor being a designated document owner.

Exception = Catastrophic, Process, Unauthorized-Access, Operand-Value  
Exception Code = X'C40309'  
Exception data = LLIDF of SOURCE-ADDRESS operand and data

- The formatting process specified in the FORMATTER-NAME operand cannot be found.

Exception = Catastrophic, Process, Data-Not-Found, Operand-Value  
 Exception Code = X'C40709'  
 Exception data = LLIDF of FORMATTER-NAME operand and data

## MODIFY

<u>Command</u>	<u>Operands</u>
MODIFY	IDENTIFIED-DATA, MODIFY-DATA [ ,ORIGINATING-NODE-ADDRESS] [ ,SOURCE-ADDRESS] [ ,SOURCE-PASSWORD]

The MODIFY command is used by authorized requestors to add or delete parameters in document related objects. The document related objects that can be modified are document access codes and search parameters defined by document profile parameters.

To modify document access codes, the requestor must be the primary owner of the document. Both primary and delegate owners are allowed to modify document search parameters.

### Operand Descriptions

#### IDENTIFIED-DATA

The IDENTIFIED-DATA operand identifies the document within the document library whose control objects are to be modified. The document may be identified by using either a Search Result List or Library Assigned Document Name (LADN). IDENTIFIED-DATA (Format 3) is used for identifying documents with the Search Result List and IDENTIFIED-DATA (Format 42) is used for identifying documents with the Library Assigned Document Name (LADN).

## MODIFY-DATA

The MODIFY-DATA (Format 41) operand is used to add or delete identified parameters within specified DIA objects. The operand specifies the following:

- DIA object identifier
- Parameter to be modified
- Operator code - add or delete
- Value to be added or deleted.

One or more sets of these data fields may be included in the operand.

The DIA object identifier field identifies the object in which the parameter to be modified is defined. This field is required and is always 3 bytes long. The 3 bytes specify the IDF introducer assigned to each particular DIA object. For example, if the parameter to be modified is a search parameter, this field specifies the IDF of the subprofile (such as the base subprofile or document library services application subprofile) in which the search parameter is defined.

The second data field identifies the parameter to be modified. This field is required and is 3 bytes long. The 3 bytes specify the IDF introducer assigned to each specific object parameter. For example, if an author search parameter is to be added, the DIA object identifier (IDF) of the base subprofile where the author parameter is defined would be specified in the first data field and the parameter (IDF) of the author profile parameter would be specified in the second data field. If the object itself is the parameter to be modified, then the encoding of the parameter data field is defined to be X'000000'. For example, the access code DIA object is itself the parameter to be modified. Therefore, the DIA object identifier field is sufficient to uniquely identify the parameter to be modified.

The third data field, the operation code, defines the operation to be performed: X'01' for adding to the parameter or X'02' for deleting from the parameter.

The fourth data field, the parameter value, contains the 1- to 245-byte data value. For add requests, the data value is appended to the DIA object parameter. For delete requests, the data value is used to locate and delete the DIA object parameter.

## ORIGINATING-NODE-ADDRESS

The ORIGINATING-NODE-ADDRESS (Format 1) operand, if present, specifies the 1- to 8-character group address token of the requestor which initiated this request. If omitted, the group address token of the command sender is assumed.

## SOURCE-ADDRESS

The SOURCE-ADDRESS operand, if present, specifies the element address token of the requestor which initiated this request; either SOURCE-ADDRESS (FORMAT 1 or 42) is valid. If omitted, the element address token of the command sender is assumed.

## SOURCE-PASSWORD

The SOURCE-PASSWORD (Format 1) operand specifies the 1- to 8-character access authorization key of the requestor which initiated this request. The SOURCE-PASSWORD (Format 1) operand is conditionally required if the SOURCE-ADDRESS (Format 1) is specified. The SOURCE-PASSWORD (Format 1) operand is required if:

- The SOURCE-ADDRESS (Format 1) operand specifies an element address token different than the command sender.
- The specified requestor does not have affinity with the command sender.
- The specified requestor has a password.

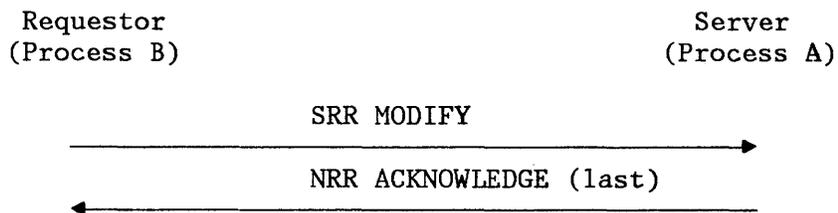
Otherwise, the SOURCE-PASSWORD should be omitted.

## Request/Reply Protocol

The following scenarios illustrate possible replies to the MODIFY command.

- Scenario 1 - Normal Completion

The replying command to a MODIFY command is an ACKNOWLEDGE command that is sent to the requestor when the specified control object parameters have been successfully added and/or deleted.



- Scenario 2 - Exception Conditions.

Exception conditions detected during the MODIFY command processing will be replied to with an ACKNOWLEDGE command that contains the exception condition in the EXCEPTION-CODE operand.

Requestor  
(Process B)

Server  
(Process A)

SRR MODIFY

NRR ACKNOWLEDGE (last)

### Exception Conditions

The general exception conditions that are common to all DIA commands are described in "DIU General Exception Conditions" in Concepts and Structures. The following exception conditions are specific to the MODIFY command and are detected and reported in addition to the general exception conditions.

- The SOURCE-PASSWORD operand is present and the SOURCE-ADDRESS (Format 1) operand is not present.

Exception = Catastrophic, Syntax, Data-Not-Found, Command-Operand  
Exception Code = X'C20708'  
Exception data = LLIDF of SOURCE-ADDRESS (Format 1) operand

- The SOURCE-ADDRESS operand specifies a requestor different than the command sender; the requestor does not have affinity with the command sender; the requestor has a password; and the SOURCE-PASSWORD operand is not specified.

Exception = Catastrophic, Semantic, Data-Not-Found, Command-Operand  
Exception Code = X'C30708'  
Exception data = LLIDF of SOURCE-PASSWORD operand

- The SOURCE-ADDRESS operand contains an address.

Exception = Catastrophic, Semantic, Data-Not-Supported, Operand-Value  
Exception Code = X'C30209'  
Exception data = LLIDF of SOURCE-ADDRESS operand and data

- The ORIGINATING-NODE-ADDRESS operand contains an invalid address.

Exception = Catastrophic, Semantic, Data-Not-Supported, Operand-Value  
Exception Code = X'C30209'  
Exception data = LLIDF of ORIGINATING-NODE-ADDRESS operand and data

- The SOURCE-PASSWORD operand contains an invalid authorization key.

Exception = Catastrophic, Semantic, Password-Invalid, Operand-Value  
Exception Code = X'C30509'  
Exception data = LLIDF of SOURCE-PASSWORD operand and data

- The document specified in the IDENTIFIED-DATA operand cannot be found.

Exception = Catastrophic, Process, Data-Not-Found, Operand-Value  
Exception Code = X'C40709'  
Exception data = LLIDF of IDENTIFIED-DATA operand and data

- The requestor is not a primary owner or delegate-owner of the document specified in the IDENTIFIED-DATA operand. The requested function cannot be executed without being a designated document owner.

Exception = Catastrophic, Process, Unauthorized-Access, Operand-Value  
Exception Code = X'C40309'  
Exception data = LLIDF of SOURCE-ADDRESS operand and data



## APPENDIX A. OPERAND DESCRIPTIONS

This section contains a detailed discussion of each operand relevant to the DIA Application Processing Services. Each discussion includes an illustration of the operand structure.

### CORRELATION (FORMAT 1)

The CORRELATION (Format 1) operand identifies the command to which this command is replying. It is used to correlate a replying command to a previously received request command. The operand is defined as follows:

LL	I	D	F	
X'nnnn'	X'C3'	X'28'	X'01'	COR
0	2	3	4	5 v

The COR operand value has the following format:

<u>FIELD</u>	<u>LENGTH</u>	<u>VALUE</u>
Reply-Indicator	1	binary
Last		X'00'
Not Last		X'01'
Reserved		X'02' - X'FF'
Command-Sequence-No.	1	binary
DIU-ID	LL-7	binary

#### Field Descriptions

The Reply-Indicator field specifies whether this reply is the last reply to the referenced request.

The Command-Sequence-Number field specifies a number which is equal to the position of the requesting command in the command sequence in the DIU in which the requesting command was received.

The DIU-ID field matches the DIU-ID field of the DIU Prefix in which the requesting command was received.

The combination of the DIU-ID and the Command-Sequence-Number parameters provide a unique identification by which the command can be correlated with the requesting command.

## FORMATTED-DOCUMENT-NAME (FORMAT 1)

The FORMATTED-DOCUMENT-NAME (Format 1) operand specifies the name of the document that is the output result of the FORMAT command. The name of the document will appear as immediate data.

LL	I	D	F	
X'nnnn'	X'C3'	X'09'	X'01'	FDN
0	2	3	4	5 v

The FDN operand value consists of 1 to 44 characters that identify the formatted document. The length may be determined from the LL bytes of the operand.

## FORMATTER-NAME (FORMAT 1)

The FORMATTER-NAME (Format 1) operand specifies the name of a format process that will be executed using the data contained in the IDENTIFIED-DATA operand as input.

LL	I	D	F	
X'nnnn'	X'C3'	X'0C'	X'01'	FNA
0	2	3	4	5 v

The FNA operand value is a 1- to 32-byte character string.

## FORMAT-PARAMETERS (FORMAT 1)

The FORMAT-PARAMETERS (Format 1) operand specifies the parameters that may be specified for the named formatter to control the formatter processing options. The syntax of the parameter data is defined by the named formatter and is preserved in the FORMAT-PARAMETERS operand exactly as it is entered.

LL	I	D	F	
X'nnnn'	X'C3'	X'37'	X'01'	FPA
0	2	3	4	5 v

The FPA operand value contains an immediate data field. The total length of the operand is specified by the LL bytes.

The FPA operand data is the exact form of the parameters as entered by the requestor. The parameter data syntax is defined by each of the formatters that may be specified by the format names supported by the server node.

## IDENTIFIED-DATA (FORMAT 1)

The IDENTIFIED-DATA (Format 1) operand specifies the location of the data being referenced by the command.

The operand is a reference to the DIU document unit. The operand value contains a 1-byte binary number designating the specific document unit from the first document unit in the DIU, that is, the nth document unit in the DIU.

LL	I	D	F	
X'nnnn'	X'C5'	X'20'	X'01'	IDD
0	2	3	4	5 v

## IDENTIFIED-DATA (FORMAT 3)

The IDENTIFIED-DATA (Format 3) operand references a document that is a member of a specified Search Result List. The operand values are specified as immediate data.

LL	I	D	F	
X'nnnn'	X'C3'	X'20'	X'03'	IDD
0	2	3	4	5 v

The IDD operand value consists of three required data fields. The first data field is the Search Result List entry number; the second data field is the type of document reference; and the third field is the Search Result List ID. These fields have the following format:

<u>FIELD NAME</u>	<u>VALUE</u>	<u>LENGTH</u>
SRL-ENTRY-NUMBER	X'0001' - X'7FFF' X'0000' & X'8000' - X'FFFF' Reserved	2 bytes
DOCUMENT-REFERENCE	X'01' Document & document profile X'02' Document content only X'03' Profile without document X'04' Document Descriptor Document X'05' Selected document descriptors and document content X'00' & X'06' - X'FF' Reserved	1 byte
SEARCH-RESULT-LIST-ID	1 to 8 characters	1 to 8 bytes

### Field Descriptions

The SRL-Entry-Number field is a 2-byte binary value of 1 to 32,767 that specifies the number of the document result entry in the Search Result List. The list entry identifies the document that is processed by the command in which this operand is specified.

The Document-Reference field specifies the type of document object to be processed for the request. This allows reference to a library document with and without its profile, reference to only the profile, reference to a Document Descriptor Document created from search-selected profile parameters, or reference to search-selected profile parameters and the document content.

The Search-Result-List-ID (SRL-ID) field is the 1- to 8-byte search request name assigned by the requestor that is the output of the SEARCH command process. The contents of the SRL-ID named object are the document references or pointers to the documents that were selected by a SEARCH command process.

### IDENTIFIED-DATA (FORMAT 42)

The IDENTIFIED-DATA (Format 42) operand specifies the Library Assigned Document Name (LADN) of the document being referenced by the command. The LADN of the document will appear as immediate data.

LL	I	D	F	LT	LT		
X'nnnn'	X'C3'	X'20'	X'42'	X'0A01'	DTM	X'nm02'	DNID
0	2	3	4	5	8		

The IDD operand value consists of two fields that uniquely identify the data being referenced. The length may be determined from the LL bytes of the operand. Each field of the IDD operand is assigned a T value that designates the field that is used to qualify the named data. The L value preceding each of the fields specifies the length of the data, including the LT bytes.

### Field Descriptions

The Library Assigned Document Name (LADN) consists of the document library node address of the document library concatenated with the date and time that the Document Library Services process completed filing and naming the document DTM.DNID. The fields that are used to generate the LADN are defined in the operand by the following field descriptions:

The DTM field of the operand specifies the date and time that the library process filed the document and created the LADN. The Date-Time field is assigned a T value of X'01'. The DTM syntax is defined as 8 bytes of discontinuous binary in the following format.

DATE-TIME ::= YYMDhms<sub>hs</sub>  
 where: YY = 2-byte binary value of 4-digit decimal year  
           (for example 1980(10): X'07BC')

M = 1-byte binary value of 2-digit decimal month  
      (for example 1-12(10): X'01' - X'0C')

D = 1-byte binary value of 2-digit decimal day of  
    month (for example 1-31(10): X'01' - X'1F')

h = 1-byte binary value of 2-digit decimal hours  
    (for example 0-23(10): X'00' - X'17')

m = 1-byte binary value of 2-digit decimal minutes  
    (for example 0-59(10): X'00' - X'3B')

s = 1-byte binary value of 2-digit decimal seconds  
    (for example 0-59(10): X'00' - X'3B')

hs = 1-byte binary value of 2-digit decimal hundredths  
      of a second (for example 0-99(10): X'00' - X'63')

The DNID field of the LADN is the node ID for the Document Library in which the named document resides. The DNID field is a 1- to 8-byte character string. The DNID field is assigned a T value of X'02'.

#### MODIFY-DATA (FORMAT 41)

The MODIFY-DATA operand specifies the identity of the DIA object parameters and the values to be used to alter the parameter contents. The operand data consists of four data fields which identify the DIA object and parameter to be modified, specify the operation code for the desired modification process, and define the modification data values. One or more sets of these data fields may be included in the MODIFY-DATA operand. Each set of data fields must be preceded by an LT byte pair specifying a 1-byte length value and a 1-byte type value.

LL	I	D	F	
X'nnnn'	X'C3'	X'10'	X'41'	MODIFY DATA
0	2	3	4	5 v

The total length of the operand is specified by the LL bytes.

#### FIELD DESCRIPTIONS

The first data field specifies the identification and format of the DIA object in which the parameters to be modified are defined. This field is required and is always 3 bytes long. The 3 bytes specify the IDF introducer assigned to each particular DIA object. For example, if the parameter to be modified is a search parameter, this field specifies the IDF of the subprofile (base subprofile, document library services application subprofile, and so forth) in which the search parameter is defined.

The second data field identifies the parameter to be modified. This field is required and is 3 bytes long. The 3 bytes specify the IDF introducer assigned to each specific object parameter.

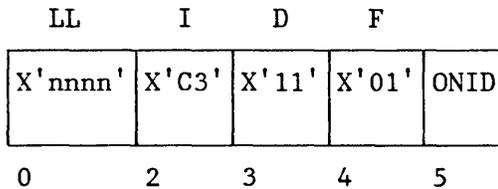
The third data field is the operation code that specifies either X'01' for adding to the parameter or X'02' for deleting from the parameter.

The fourth data field contains either the data value that will be added to the identified parameter or that is used to locate the data value of the identified parameter field to be deleted. The Modify-Data-Value field must include the LT identifier for all parameter data field modifications when they are specified as part of the DIA object parameter definition.

<u>LENGTH</u>	<u>TYPE</u>	<u>FIELD NAME</u>	<u>VALUE</u>
X'nn'	X'01'	MODIFY-DATA-INTRODUCER	Length and T byte introducer. The length value is total length of the modify data, including the L and T bytes.
X'iiddff'		OBJECT-INTRODUCER	The assigned IDF value for the DIA object to be modified.
X'iiddff'		PARAMETER-INTRODUCER	The assigned IDF value for the parameter to be modified. This field may be X'000000' when a parameter introducer is not used.
X'nn'		MODIFY-OPERATION-CODE	X'01' Add modify data value to identified parameter. X'02' Delete modify data value from identified parameter if present. X'00' reserved X'03' - X'FF' reserved
		MODIFY-DATA-VALUE	1- to 245-byte character string to be used to modify the identified parameter. LT identifiers must be included when defined in the parameter.

## ORIGINATING-NODE-ADDRESS (FORMAT 1)

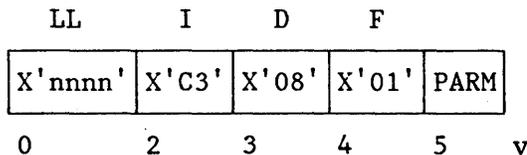
The ORIGINATING-NODE-ADDRESS (Format 1) operand identifies the group address token of the command requestor.



The group address token specified by the ONID operand value is the node address of the OSN that originated the function to be performed in support of the command containing this operand. ONID is a 1- to 8-byte character string.

## PROCESS-PARAMETERS (FORMAT 1)

The PROCESS-PARAMETERS (Format 1) operand specifies the parameters that may be specified for the named process of the EXECUTE command to control execution options. The syntax of the parameter data is defined by the named process and is preserved in the operand exactly as it is entered.

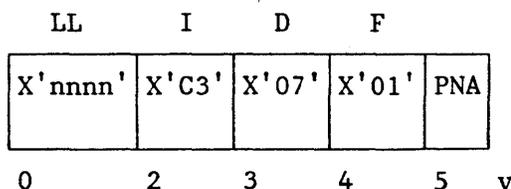


The PARM operand value contains an immediate data field. The total length of the operand is specified by the LL bytes.

The PARM operand data is the exact form of the parameters as entered by the requestor. The parameter data syntax is defined by each of the processes that may be specified by the process names supported by the server.

## PROCESS-NAME (FORMAT 1)

The PROCESS-NAME (Format 1) operand specifies the name of the process, program, or procedure which is to be executed in the operating system of the server.



The PNA operand value is a 1- to 32-byte character string.

## PROCESS-PASSWORD (FORMAT 1)

The PROCESS-PASSWORD (Format 1) operand specifies the process authorization key that permits executing the named process.

LL	I	D	F	
X'nnnn'	X'C3'	X'13'	X'01'	PPWD
0	2	3	4	5 v

The PPWD operand value is a 1- to 8-byte character string.

## SOURCE-ADDRESS (FORMAT 1)

The SOURCE-ADDRESS (Format 1) operand specifies the element address token of the requestor source node.

LL	I	D	F	
X'nnnn'	X'C3'	X'23'	X'01'	SID
0	2	3	4	5

The element address token specified by the SID operand value is the requestor's source node address and is used at the application level to identify the user or process that is the source of the DIA command and its related data. SID is a 1- to 8-byte character string.

## SOURCE-ADDRESS (FORMAT 42)

The SOURCE-ADDRESS (Format 42) operand specifies the element address token of the requestor source node.

LL	I	D	F	LT		LT	
X'nnnn'	X'C3'	X'23'	X'42'	X'nn01'	DOMID	X'nn02'	SN
0	2	3	4	5			

	LT		LT
X'nn03'	GN	X'nn04'	AV

The DOMID part of this operand specifies the domain ID and is used at the application level to partially identify the user or process that is the

requestor of the DIA command and its related data. The domain ID is unique within the ONID specified by the ORIGINATING-NODE-ADDRESS operand that is associated with this SOURCE-ADDRESS operand. If there is no associated ORIGINATING-NODE-ADDRESS operand, then DOMID must be unique within the OSN where it is received. DOMID is a 1- to 8-byte character string.

The SN operand value specifies the source name and is also used at the application level to further identify the user or process. The source name is unique within the domain specified by the domain ID in this SOURCE-ADDRESS operand. SN is a 1- to 32-byte character string.

The GN part of this operand specifies the global name that is associated with the requestor. GN is a 1- to 32-byte character string.

The AV operand value is an authorization value that is associated with the user or process identified by either the SN or the GN value. The AV part of this operand may appear only when this operand is being used in a command from a source node to an originating office system node. AV is a 1- to 8-byte character string.

The L byte for each of these operand parts specifies the length of each construct including the two LT bytes and the 1- to 8- or 1- to 32-byte character string.

When Format 42 of the SOURCE-ADDRESS operand is used, it must contain one of the following combinations of the individual operand parts. All parts specified for a given combination must be present.

Domain ID, Source Name, and Authorization Value

Domain ID and Global Name

Domain ID, Global Name, and Authorization Value

## SOURCE-PASSWORD (FORMAT 1)

The SOURCE-PASSWORD (Format 1) operand is an access authorization key associated with a requestor.

LL	I	D	F		
X'nnnn'	X'C3'	X'0E'	X'01'	SP	
0	2	3	4	5	v

The SP operand value is a 1- to 8-byte character string.



## APPENDIX B. DOCUMENT INTERCHANGE UNIT (DIU)

The basic unit of interchange exchanged between DIA processes is the document interchange unit (DIU). A DIU is made up of the following data stream components:

PREFIX	COMMAND SEQUENCE	DATA UNITS	DOCUMENT UNITS	SUFFIX
--------	---------------------	---------------	-------------------	--------

- The prefix contains the information to introduce and identify the DIU.
- The command sequence contains the command that specifies the function to be performed and related processing information.
- The data unit contains information that may be referred to by the DIA command in the command sequence. This field is optional and is present when defined by the command.
- The document unit contains the document profile and optionally the document content. This field is optional and is present only when a document profile and content are sent from one DIA process to another.
- A suffix specifies the end of the DIU and indicates whether any abnormal conditions occurred while the DIU was being transmitted.

These data stream components may be composed of substructures called subcomponents. Examples of subcomponents are command operands and document profiles. All DIU components and their subcomponents begin with a structured field called an introducer. The introducer uniquely identifies each field and indicates its length. Consequently all fields and components (and hence, the entire data stream) are self describing and may be variable in length.

### DIU STRUCTURED FIELD

The DIU structured field consists of four parts; the total length (LL) of the structured field, the structured field identifier (IDF), an optional structured field identifier extension (ISS), and an optional data variable.

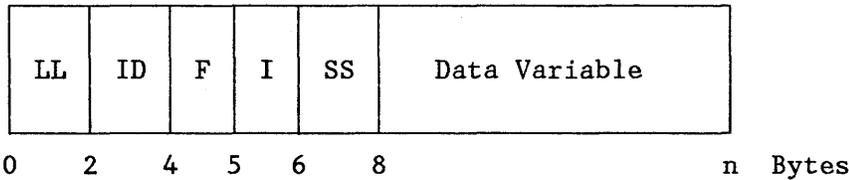


Figure 2. DIU Structured Field

DIU Introducer

The LLIDF part of the DIU structured field is called the DIU Introducer (INTRO). All DIU data stream components and subcomponents contain an INTRO (LLIDF).

The optional ISS part of the structured field is called the DIU Introducer Extension. Introducer extensions (ISS) are permitted on major DIU data stream components only: prefix, commands, data units, document units, and suffix.

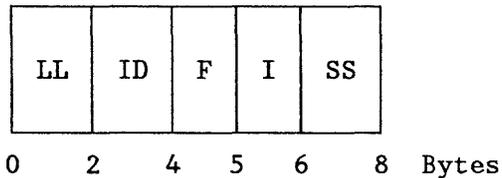


Figure 3. DIU Introducer (INTRO)

LL = Structured field length

The length LL may vary from 5 to 32,767 bytes; including the LLIDF(ISS) and structured field data variable.

ID = Structured field identifier.

The ID consists of two parts; a class byte and type byte where:

- I = Construct class (for example, Prefix, Command, Operand)
- D = Construct type (for example, Type of command)

F = Format byte

The Format byte defines the format of the data variable and indicates whether the optional ISS is present.

The F byte is defined as follows:

- Bit 0 - Introducer Extension Indicator
  - 1 = ISS is present and follows the LLIDF
  - 0 = No ISS is present
- Bit 1 - Imbedded Structure Indicator

- 1 = Data variable format is in LT format
- 0 = Data variable is defined by Bits 4-7
- Bits 2 - 3 Reserved
- Bits 4 - 7 Data Variable Format Indicator
- A 4-bit binary number specifying the format and syntax of the data variable.

DIU Introducer Extension (ISS)

The ISS portion of the structured field is defined as follows:

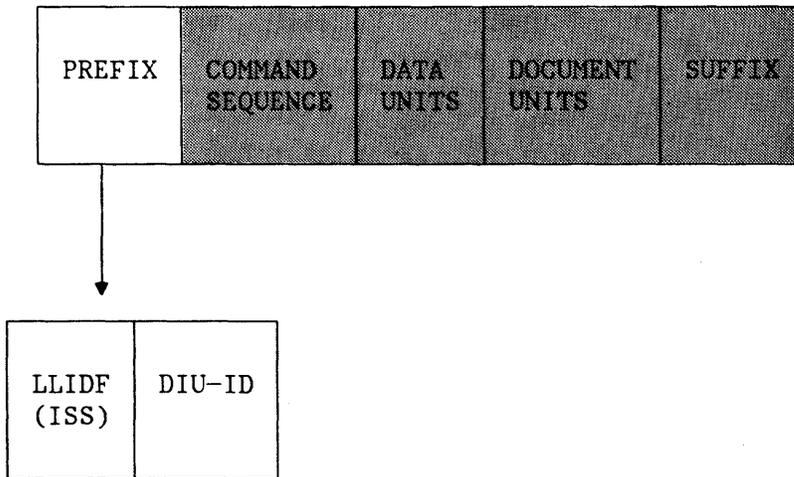
I = Indicator byte

The Indicator byte identifies the structure of the construct.

- Bits 0 - 1 Reserved
- Bit 2 - Segmentation Indicator
- 1 = Not last, a structured field segment follows
- 0 = Last or only structured field segment
- Bits 3 - 7 Reserved

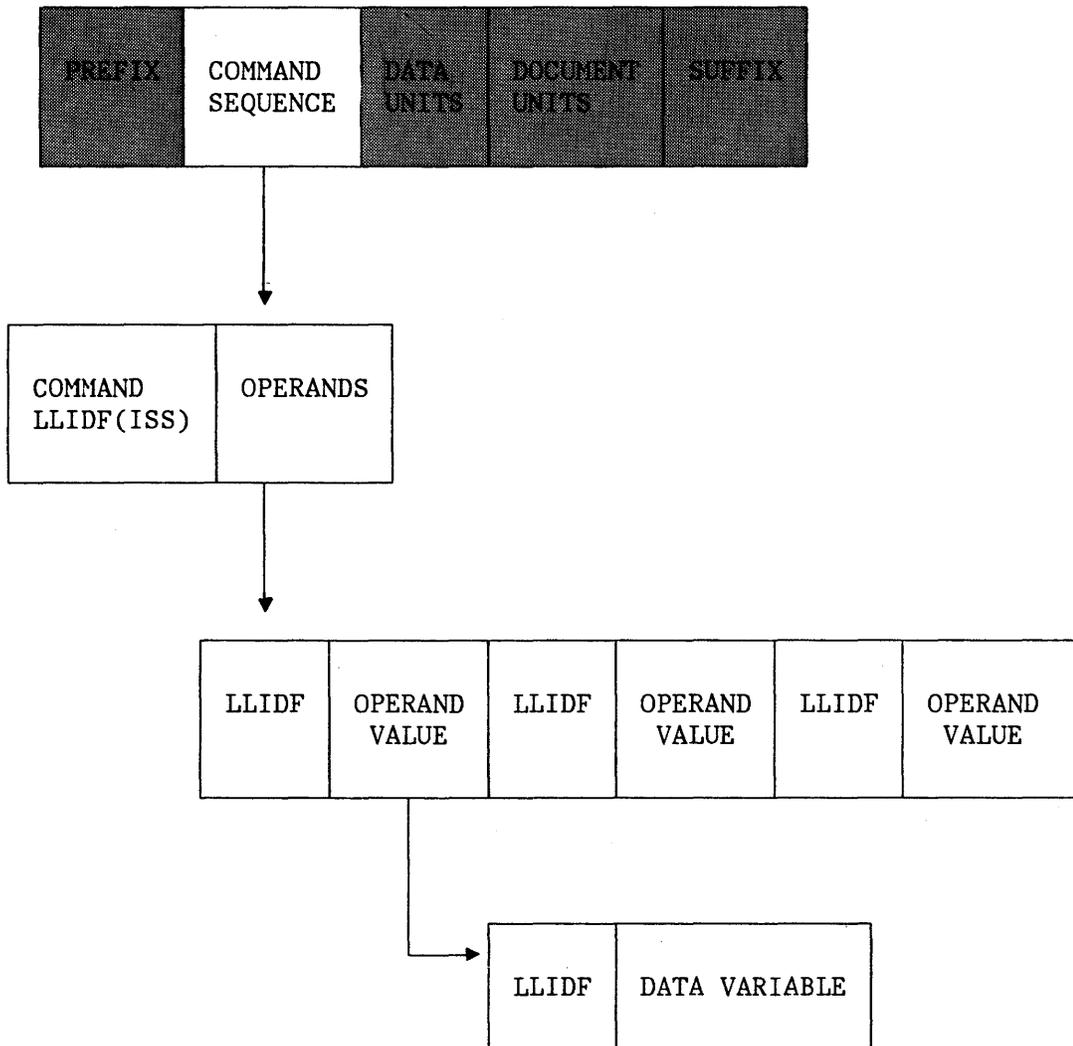
SS = Sequence number - X'0000'

**DIU PREFIX**



The DIU prefix contains a 0- to 16-byte value called a DIU-ID. This DIU identifier, specified by the DIU sender, is used to correlate a command reply with the command request.

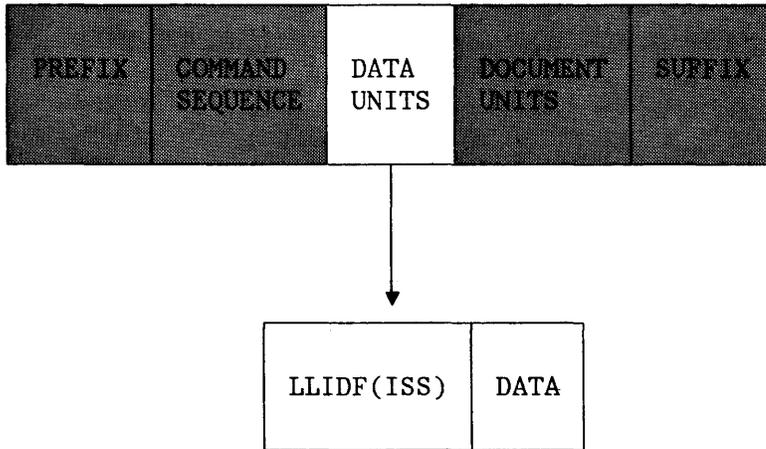
## COMMAND SEQUENCE



The COMMAND LLIDF(ISS) identifies the command that specifies the function to be performed. The command introducer contains a length field which spans the command introducer and all subordinate operands.

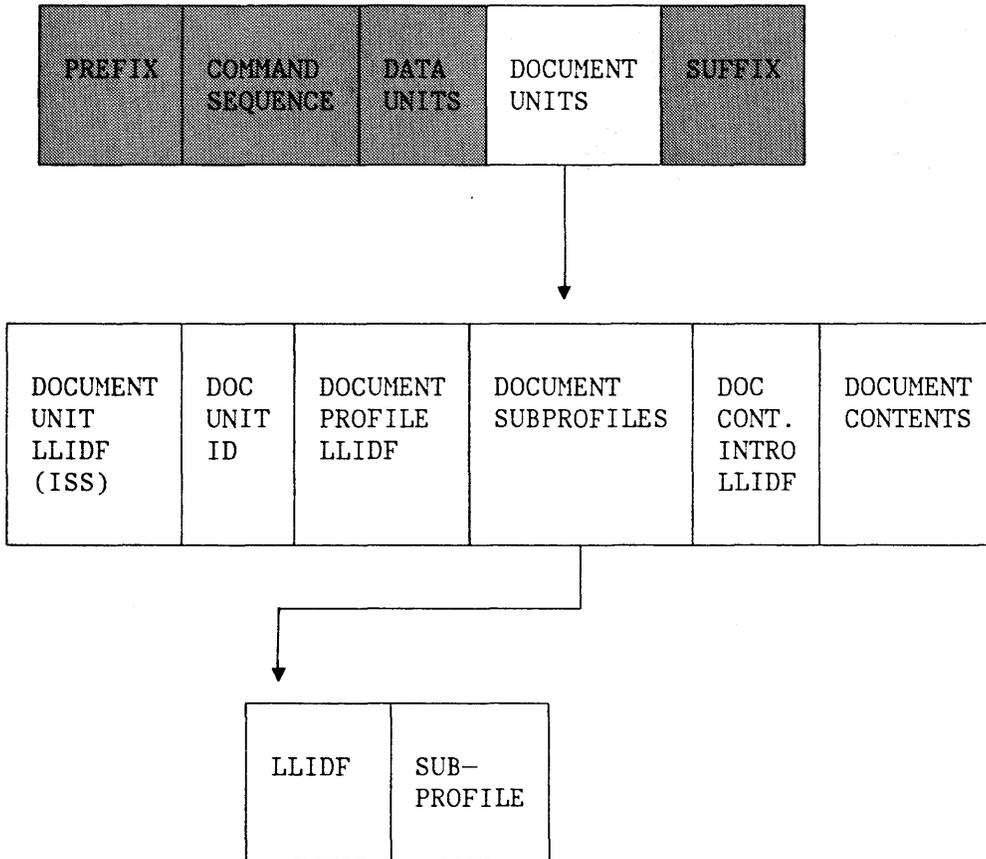
OPERANDS define input parameters to the command. Operand introducers may not contain introducer extensions. The length field of the operand introducer spans both the introducer and data variable.

## DATA UNITS



The data field contains either data that can be referenced by commands or information passed to the requestor of a command.

## DOCUMENT UNITS



The DOCUMENT UNIT INTRODUCER identifies the document unit type. The length field of the document unit introducer spans the document unit introducer, document profiles, document content introducer, and document content.

The DOC UNIT ID is a 15-byte document identifier which specifies the document type (2-byte binary number). The document types are listed in "Appendix D. DIA Document Types" on page 51.

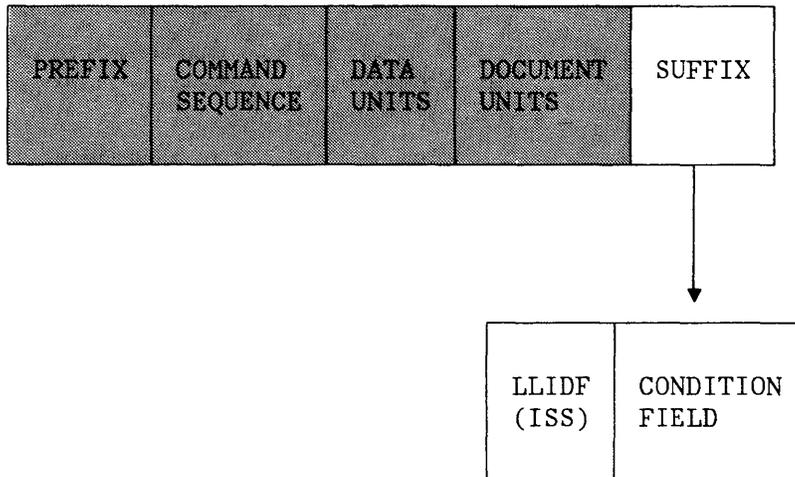
The document profile contains information relating to or describing a document, such as author or document name. The length field of an unsegmented DOCUMENT PROFILE LLIDF spans the introducers and any subordinate document subprofiles.

The introducer for each subprofile, LLIDF, contains a length field and subprofile identifier. The length field spans the introducer and subprofile information. The identifier specifies the subprofile (such as base subprofile or application subprofile) that is to follow.

The DOC CONT. INTRO identifies the beginning of the document content.

The DOCUMENT CONTENTS contain the text of the information. The document content may be defined externally from DIA or be a DIA-defined document.

## DIU SUFFIX



There are two types of DIU suffixes:

Type 1 Suffix - Normal termination of a DIU

Type 2 Suffix - Abnormal termination of a DIU

If the DIU has terminated abnormally, the CONDITION FIELD will contain an exception code describing the error which caused the termination.

If the DIU terminates normally, the CONDITION FIELD is null.



## APPENDIX C. DIA CODE POINTS

This appendix consists of tables which list the code points defined for the DIA constructs relevant to Application Processing Services. Each table gives the description of the construct and the code point which corresponds to the structured field identifier, that is, the IDF. The I corresponds to the construct class and the D corresponds to the construct type. Any code point not listed in one of the tables is reserved.

The code point tables are grouped by construct class (such as Prefix, SRR Command, or Immediate Data Operand) according to the following list:

CLASS	CODE POINT CLASS	TABLE
DIU PREFIX	X'CO'	1
COMMAND - NO REPLY REQUIRED	X'C1'	2
OPERAND - IMMEDIATE	X'C3'	4
OPERAND - DOCUMENT UNIT REFERENCE	X'C5'	5
DOCUMENT UNIT	X'C9'	6
DOCUMENT PROFILE	X'CA'	7
DOCUMENT CONTENT INTRODUCER	X'CB'	8
COMMAND - SYNC REPLY REQUIRED	X'CD'	3
DIU SUFFIX	X'CF'	9

### NOTES FOR DIA CODE POINT ASSIGNMENT TABLES

All TYPE CODES of X'00' are RESERVED.

A lower case x is used to indicate the value for bits 0 - 3 of the format byte. The actual setting of these bits must be in accordance with the rules for the specific construct in which the bits appear.

The LT form of the data field is only valid in operands and document profile parameters.

Format is always X'x0' when there is no value field, that is, LL = X'05' or X'08'.

TABLE 1. DIU PREFIX ENCODINGS

CODE POINT DESCRIPTION	GDS ID	F
DIU PREFIX	C0	
DIU PREFIX, INTERCHANGE FORM	C001	x2

TABLE 2. NO REPLY REQUIRED COMMAND ENCODINGS

CODE POINT DESCRIPTION	GDS ID	F
DIU COMMAND - NO-REPLY-REQUIRED (NRR)	C1	
NRR - ACKNOWLEDGE	C101	x1
NRR - SIGN-ON	C10C	x1
NRR - SIGN-OFF	C10D	x1
NRR - DELIVER	C119	x1

TABLE 3. SYNCHRONOUS REPLY REQUIRED COMMAND ENCODINGS

CODE POINT DESCRIPTION	GDS ID	F
DIU COMMAND - SYNCHRONOUS REPLY RQRD (SRR)	CD	
SRR - FORMAT	CDOA	x1
SRR - EXECUTE	CDOB	x1
SRR - SIGN-ON	CDOC	x1
SRR - MODIFY	CD12	x1

TABLE 4. IMMEDIATE DATA OPERAND ENCODINGS

CODE POINT DESCRIPTION	GDS ID	F
IMMEDIATE DATA OPERAND (IMMED DATA OPND)	C3	
IMMED DATA OPND PROCESS-NAME	C307	01
IMMED DATA OPND PROCESS-PARAMETERS	C308	01
IMMED DATA OPND FORMATTED-DOCUMENT-NAME	C309	01
IMMED DATA OPND MODIFY-PROFILE-DATA	C30B	01
IMMED DATA OPND FORMATTER-NAME	C30C	01
IMMED DATA OPND SIGN-ON-ID	C30D	01
IMMED DATA OPND SIGN-ON-ID	C30D	42
IMMED DATA OPND SOURCE/RECIPIENT-PASSWORD	C30E	01
IMMED DATA OPND CHARGE-CODE	C30F	01
IMMED DATA OPND MODIFY-CONTROL-DATA	C310	01
IMMED DATA OPND ORIGINATING-NODE-ADDRESS	C311	01
IMMED DATA OPND FUNCTION-SET	C312	01
IMMED DATA OPND PROCESS-PASSWORD	C313	01
IMMED DATA OPND IDENTIFIED-DATA	C320	03
IMMED DATA OPND IDENTIFIED-DATA	C320	42
IMMED DATA OPND EXCEPTION-CODE	C322	01
IMMED DATA OPND SOURCE-ADDRESS	C323	01
IMMED DATA OPND SOURCE-ADDRESS	C323	42
IMMED DATA OPND RECOVERY-ACTION	C327	01
IMMED DATA OPND CORRELATION	C328	01
IMMED DATA OPND DOCUMENT-TYPE	C329	01
IMMED DATA OPND GRAPHIC-CHARACTER-SET-ID	C32A	01
IMMED DATA OPND FORMAT-PARAMETERS	C337	01
IMMED DATA OPND SIGN-ON-PASSWORD	C338	01
IMMED DATA OPND COUNT	C33E	01
IMMED DATA OPND REPLY-DATA	C345	01

TABLE 5. DOCUMENT UNIT REFERENCE OPERAND ENCODINGS

CODE POINT DESCRIPTION	GDS ID	F
DOCUMENT UNIT REF OPND (DOC U REF OPND)	C5	
DOC U REF OPND IDENTIFIED-DATA	C520	01

TABLE 6. DOCUMENT UNIT ENCODINGS

CODE POINT DESCRIPTION	GDS ID	F
DIU DOCUMENT UNIT	C9	
DOCUMENT UNIT, INTERCHANGE	C903	x1
DOCUMENT UNIT, DOCUMENT-DESCRIPTOR-PARMS	C904	01
DOCUMENT UNIT, UNFORMATTED-RECIP'NT STATUS	C905	01
DOCUMENT UNIT, UNFORMATTED-SUMMARY STATUS	C906	01
DOCUMENT UNIT, UNFORMATTED-SOURCE STATUS	C907	01
NOTE: DOCUMENT UNIT CODES X'C980'-X'C9FF' USER ASSIGNED		

TABLE 7. DOCUMENT PROFILE ENCODINGS

CODE POINT DESCRIPTION	GDS ID	T
DIU DOCUMENT PROFILE	CA	
DOCUMENT PROFILE, PRIVATE, 3730	CA01	01
DOCUMENT PROFILE, PRIVATE, DISOSS	CA01	02
DOCUMENT PROFILE, PRIVATE (5520)	CA02	01
DOCUMENT PROFILE, INTERCHANGE (IDPA)	CA03	01
BASE SUBPROFILE (IDPA)	CA04	01
ARCHITECTED APPLICATION SUBPROFILE (DIA)	CA05	01
IBM 3730 SUBPROFILE - 3730	CA70	01
DISOSS SUBPROFILE - DISOSS	CA71	01
IBM 5520 SUBPROFILE - 5520	CA72	01
NOTE: DOCUMENT PROFILE CODES X'CA80'-X'CAFF' USER ASSIGNED		

TABLE 8. DOCUMENT CONTENT INTRODUCER ENCODINGS

CODE POINT DESCRIPTION	GDS ID	F
DIU DOCUMENT CONTENT INTRODUCER	CB	
DOCUMENT CONTENT INTRODUCER, W/DOCUMENT	CB01	01
DOCUMENT CONTENT INTRODUCER, WO/DOCUMENT	CB02	01

TABLE 9. DIU SUFFIX ENCODINGS

CODE POINT DESCRIPTION	GDS ID	F
DIU SUFFIX	CF	
DIU SUFFIX, NORMAL TERMINATION	CF01	x0
DIU SUFFIX, ABNORMAL TERMINATION	CF02	x1



## APPENDIX D. DIA DOCUMENT TYPES

The following table lists the document types registered by Document Interchange Architecture.

Interchange Data Stream Type	Identifier Code
Reserved	X'0001'
Final-Form-Text Document	X'0002'
5520 Revisable-Form-Text Document	X'0003'
Word-Processing EBCDIC	X'0004'
Word-Processing-Information-File (WPIF)	X'0005'
Image-Data-Subset Document	X'0006'
3730 Text Data Stream	X'0007'
DIA Document Library Document Descriptor Document	X'0008'
3732 Display Document Data stream	X'0009'
DIA Defined Document Unit Content	X'000A'
Revisable-Form-Text Document	X'000B'
1403 Printer Compatible Data Stream with Variable Length, Unblocked Records.	X'000C'

Figure 4. Document Type Code Assignments



## GLOSSARY

**access code.** A 4-byte decimal value, assigned to a document by the primary owner, that determines the set of users allowed to access the document.

**address.** (1) A character or group of characters that identifies a register, a particular part of storage, or some other data source or destination. (2) In DIA, a 1- to 8-byte character string that identifies the logical components of an office system network. These logical components are: source nodes, recipient nodes, and office system nodes.

**affinity.** A defined relationship that permits the DIA resources of a source or recipient to be accessed on his behalf by another user.

**application processing services.** The set of services that provide DIA functions enabling users to access processing capabilities of a remote node.

**ARR.** Asynchronous reply required.

**asynchronous reply required (ARR).** A command class that requests asynchronous processing and reply of a DIA function.

**COD.** Confirmation-of-delivery.

**command.** The function to be performed by the receiving DIA process.

**command sequence.** A DIU data stream component containing a set of one or more commands.

**condition code.** Defines the specific exception condition detected by the receiver of a DIU.

**confirmation-of-delivery (COD).** An asynchronous message returned to the source node of a distribution request that indicates the information distributed has been delivered to the recipient node.

**control variable.** A DIA entity maintained by a DIA process for the purpose of verification and authorization.

**correlation value.** Information used to uniquely identify and correlate the request to the reply.

**data unit.** A DIU data stream component that contains information referenced by operands of a command in the DIU.

**data variable.** A variable length collection of information contained in a structured field.

**destination node.** The office system node that provides services for attached source and recipient nodes.

**DIA.** Document interchange architecture.

**DIA session.** A logical connection between two DIA processes that is used to exchange information.

**distribution.** In general, the function provided by DIA of transporting information from a

source node to one or more recipient nodes.

**distribution document name.** A unique identifier assigned to each distribution request.

**distribution library.** The collection of distribution queues and data storage provided by an office system node for the purpose of document distribution.

**distribution queue.** A queue of distribution and status information to be delivered to source or recipient nodes.

**distribution system.** The collection of office system nodes, source nodes, and recipient nodes that are interconnected to form an office system network.

**DIU.** Document interchange unit.

**DIU component.** A self-defining, variable length structured field. The DIU components are: prefix, command sequence, data unit, document unit, and suffix.

**DIU subcomponent.** A self-defining, variable length structured field contained within a DIU component.

**document.** (1) (ISO) A data medium and the data recorded on it, that generally has permanence and that can be read by man or machine. (2) A unified collection of information pertaining to a specific subject or related subjects.

**document content introducer.** The DIU data stream subcomponent that identifies the beginning of the document content.

**document descriptor.** A set of profile parameters describing a

document that satisfied a document library search request.

**document descriptor document.** A collection of one or more document descriptors.

**document distribution services.** The set of services that provide DIA functions enabling users to distribute information in a distribution system.

**Document Interchange Architecture (DIA).** The specification of rules and data streams necessary to interchange information in a consistent, predictable manner.

**document interchange unit (DIU).** The basic unit of information exchanged between DIA processes.

**document library.** A repository on which documents and document related information is stored.

**document library services.** The set of services that provide DIA functions enabling users to manage the contents of a document library.

**document type.** A classification that identifies the structure and format of a document.

**document unit.** A DIU data stream component that contains the document and related document information.

**document unit identifier.** The DIU data stream subcomponent that contains the document type and system code identifier of the document.

**end user.** (1) The ultimate source or destination of information flowing through a

system. (2) In DIA, a program, device, person, or system that uses DIA for the purpose of information interchange.

**exception condition class.** The type of exception condition detected by the receiver of a DIU. The exception classes are: session, syntax, semantic, process, and sender.

**exception condition data.** A field containing the DIU data stream component or subcomponent that caused the exception condition.

**exception condition object.** An identifier of the DIU component or subcomponent that caused the exception condition.

**format byte.** That part of the structure field introducer that defines the format and content of the structured field data variable.

**function set.** The set of commands that identify the scope of work. Function sets have been defined so that each set contains all commands required for a well-defined, usable, and complete set of functions for a given category of services.

**GCID.** Graphic character set ID.

**graphic character set ID (GCID).** The registry for graphic character sets and code pages.

**ID.** That part of the structured field introducer that defines the class and type of the structured field.

**IDP.** Interchange document profile.

**Interchange Document Profile (IDP).** A set of descriptors that identify and describe a document.

**introducer.** A 5-byte structured field identifier. The introducer contains a 2-byte length field, a 2-byte ID, and a format byte.

**introducer extension.** An optional extension to the structured field introducer used for segmentation of the structured field.

**ISS.** Introducer extension.

**LADN.** Library assigned document name.

**library assigned document name (LADN).** A unique name assigned to documents filed in the document library.

**message.** A collection of information transmitted from one point to another.

**No reply required (NRR).** A command class used when the function requested does not require a reply.

**NRR.** No reply required.

**office system node.** The DIA process that provides the services for attached source or recipient nodes.

**operand.** (1) (ISO) An entity to which an operation is applied. (2) A data stream subcomponent that controls the execution of the command.

**originating node.** The office system node that provides services for attached source nodes.

**OSN.** Office system node.

**owner-delegate.** A user that is designated as secondary owner by the primary owner of the document in the document library.

**password.** A character string used for validation and authorization to gain access to a resource.

**personal.** A distribution class of service that requires the recipient to supply a password to receive the distributed information.

**prefix.** The DIU data stream component that introduces and identifies the DIU.

**primary owner.** The user who files the document in the document library.

**priority.** A distribution class of service that prioritizes the distributions so information of higher priority is delivered before information of lower priority.

**process.** (1) A systematic sequence of operations to produce a specified result. (2) In DIA, a program that uses the DIA rules and data structures to interchange information.

**profile parameter.** A field of a subprofile that identifies and describes the document.

**recipient.** An end user that receives information in an office system network.

**recipient node.** A DIA logical component that provides services on behalf of recipients.

**recovery action.** The procedure recommended by the process that detected an exception condition.

**reply.** A command that is used to respond to a previously received request.

**request.** A command that specifies a function to be performed.

**search argument.** A search selection criterion that contains the profile parameter identifier, the search data value, and the search comparison operator.

**search data parameter set.** A collection of one or more search data parameters and the logical operators used to relate them.

**search result list.** A user named object that contains references to documents selected by the SEARCH command process.

**segmentation.** The division of a DIU data stream component into two or more segments.

**source.** An end user that requests services in an office system network.

**source node.** A DIA logical component that provides services on behalf of sources.

**SRR.** Synchronous reply required.

**structured field.** A self-defining, variable length field comprised of an introducer, an optional introducer extension, and a data variable.

**subprofile.** A set of profile parameters that describe the characteristics and attributes of a document.

**suffix.** The DIU data stream component that terminates the DIU.

**synchronous reply required (SRR).** A command class that requests synchronous processing and reply of a DIA function.

**system code.** An identifier associated with the originator of the document that is contained in a DIU document unit.

**user.** See end user.



## INDEX

### A

access codes 21  
ACKNOWLEDGE command 5, 6, 9, 11,  
14, 15, 18, 19, 23  
application processing  
services 1, 4, 7, 9  
basic concepts of 4

### C

CORRELATION (Format 1) operand 5,  
6, 27

### D

delegate-owner 21  
DELIVER command 7, 9-13, 17, 18  
description 9  
exception conditions  
process 12  
semantic 11-12  
session 11  
syntax 11  
operands  
CORRELATION 10  
IDENTIFIED-DATA 10  
protocols  
exception conditions 11  
multiple replies 10  
single reply 10  
destination node address 4  
DIA command 6  
correlation of 5, 6  
general protocol rules 6  
DIA command class 5  
asynchronous reply required  
(ARR) 6  
no reply required (NRR) 5  
synchronous reply required  
(SRR) 6  
DIA session 6, 7  
DIU identifier 6  
document descriptors 1, 4  
Document Distribution Services 1,  
13

### Document Interchange

Architecture 1, 3, 7  
See also application processing  
services  
See also Document Distribution  
Services  
See also Document Library  
Services  
basic concepts of 3  
categories of services 1  
command 3  
process 3, 5, 6, 7  
relationship to SNA 3  
services defined by 1  
services performed by 3  
document interchange unit  
(DIU) 4, 5, 6  
components 4, 6  
command sequence 4  
data unit 4  
document unit 4  
prefix 4, 6  
suffix 4  
flow between processes 6  
subcomponents 5  
document library 1, 3  
Document Library Services 1, 18,  
30  
document profile 4, 5

### E

end user 3, 4  
EXCEPTION-CODE operand 11, 15,  
18, 19, 23  
EXECUTE command 7, 13-16  
description 13  
exception conditions  
process 16  
semantic 15-16  
syntax 15  
operands  
IDENTIFIED-DATA 13  
ORIGINATING-NODE-ADDRESS 14  
PROCESS-NAME 13  
PROCESS-PARAMETERS 14  
PROCESS-PASSWORD 14  
SOURCE-ADDRESS 14

SOURCE-PASSWORD 14  
protocols  
exception condition 15  
normal completion 14

## F

FILE command 18  
final-form-text 1, 4  
FORMAT command 7, 9, 16-21  
description 16  
exception conditions  
process 20-21  
semantic 20  
syntax 20  
operands  
FORMAT-PARAMETERS 18  
FORMATTED-DOCUMENT-NAME 17  
FORMATTER-NAME 17  
IDENTIFIED-DATA 17  
ORIGINATING-NODE-ADDRESS 17  
SOURCE-ADDRESS 17  
SOURCE-PASSWORD 17  
protocols  
exception condition 19  
formatted data filed 18  
formatted data returned 18  
FORMAT-PARAMETERS (Format 1)  
operand 28  
FORMATTED-DOCUMENT-NAME (Format 1)  
operand 28  
FORMATTER-NAME (Format 1)  
operand 28  
function set 7  
negotiation of 7  
reason for 7

IDENTIFIED-DATA (Format 1)  
operand 29  
IDENTIFIED-DATA (Format 3)  
operand 29-30  
IDENTIFIED-DATA (Format 42)  
operand 30-31

## L

LADN 18, 21, 30, 31  
description 30  
Library Assigned Document Name  
See LADN  
library, document 1

## M

MODIFY command 7, 21-25  
description 21  
exception conditions  
process 25  
semantic 24  
syntax 24  
operands  
IDENTIFIED-DATA 21  
MODIFY-DATA 22  
ORIGINATING-NODE-ADDRESS 22  
SOURCE-ADDRESS 23  
Source-Password 23  
protocols  
exception condition 23  
normal completion 23  
MODIFY-DATA (Format 41)  
operand 31-32

## O

office system network 3  
logical components of 3  
office system node 3, 4, 7  
destination OSN functions 4  
end user functions 4  
originating OSN functions 4  
originating node address 4  
ORIGINATING-NODE-ADDRESS  
(Format 1) operand 33  
OSN  
See office system node

## P

prerequisite publication iii  
primary owner 18, 21  
process A 7  
process B 7  
PROCESS-NAME (Format 1)  
operand 33

PROCESS-PARAMETERS (Format 1)  
  operand 33  
PROCESS-PASSWORD (Format 1)  
  operand 34  
publication, prerequisite iii  
publications, related iii

## R

recipient address 4  
recipient node 3  
related publications iii  
requestor node 7  
revisable-form-text 1, 4

## S

SEARCH command 30  
Search Result List  
  See SRL  
server node 7  
SIGN-OFF command 9

SIGN-ON command 7, 9  
source address 4  
source node 3, 4  
  end user functions 4  
SOURCE-ADDRESS (Format 1)  
  operand 34  
SOURCE-ADDRESS (Format 42)  
  operand 34-35  
SOURCE-PASSWORD (Format 1)  
  operand 35  
SRL 29, 30  
structured field introducer 5



You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

*Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comment are:

Clarity      Accuracy      Completeness      Organization      Coding      Retrieval      Legibility

If you wish a reply, give your name, company, mailing address, and date:

---

---

---

---

Note: Staples can cause problems with automated mail sorting equipment.  
Please use pressure sensitive or other gummed tape to seal this form.

What is your occupation? \_\_\_\_\_

Number of latest Newsletter associated with this publication: \_\_\_\_\_

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Reader's Comment Form

• • • Cut or Fold Along Line • • •

Fold and tape

Please Do Not Staple

Fold and tape

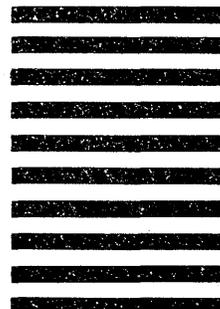


NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
Department D31, Building 803  
11400 Burnet Rd.  
Austin, Texas 78758



Fold and tape

Please Do Not Staple

Fold and tape

Document Interchange Architecture: Application Processing Services Reference Printed in U.S.A. SC23-0761-0



