ASG
Software Solutions

ASG-DataManager™
User Interface Facility

Version 2.5.1
Publication Number: DMR0200-251-UI
Publication Date: November 2001

The information contained herein is the confidential and proprietary information of Allen Systems Group, Inc. Unauthorized use of this information and disclosure to third parties is expressly prohibited. This technical publication may not be reproduced in whole or in part, by any means, without the express written consent of Allen Systems Group, Inc.

© 1998-2002 Allen Systems Group, Inc. All rights reserved.
All names and products contained herein are the trademarks or registered trademarks of their respective holders.

ASG Worldwide Headquarters Naples Florida USA | asg.com | info@asg.com
1333 Third Avenue South, Naples, Florida 34102 USA Tel: 239.435.2200 Fax: 239.263.3692 Toll Free: 800.932.5536 (USA only)

ASG Documentation/Product Enhancement Fax Form

Please FAX comments regarding ASG products and/or documentation to (239) 263-3692.

Company Name	Telephone Number	Site ID	Contact name

Product Name/Publication	Version #	Publication Date
Product:		
Publication:		
Tape VOLSER:		

Enhancement Request:

© 2002 Allen Systems Group, Inc.
All names and products are trademarks or registered trademarks of their respective holders.

ASG Support Numbers

ASG provides support throughout the world to resolve questions or problems regarding installation, operation, or use of our products. We provide all levels of support during normal business hours and emergency support during non-business hours. To expedite response time, please follow these procedures.

Please have this information ready:

- Product name, version number, and release number
- List of any fixes currently applied
- Any alphanumeric error codes or messages written precisely or displayed
- A description of the specific steps that immediately preceded the problem
- The severity code (ASG Support uses an escalated severity system to prioritize service to our clients. The severity codes and their meanings are listed below.)
- Verify whether you received an ASG Service Pack for this product. It may include information to help you resolve questions regarding installation of this ASG product. The Service Pack instructions are in a text file on the distribution media included with the Service Pack.

If You Receive a Voice Mail Message:

1. Follow the instructions to report a production-down or critical problem.
2. Leave a detailed message including your name and phone number. A Support representative will be paged and will return your call as soon as possible.
3. Please have the information described above ready for when you are contacted by the Support representative.

Severity Codes and Expected Support Response Times

Severity	Meaning	Expected Support Response Time
1	Production down, critical situation	Within 30 minutes
2	Major component of product disabled	Within 2 hours
3	Problem with the product, but customer has work-around solution	Within 4 hours
4	"How-to" questions and enhancement requests	Within 4 hours

ASG provides software products that run in a number of third-party vendor environments. Support for all non-ASG products is the responsibility of the respective vendor. In the event a vendor discontinues support for a hardware and/or software product, ASG cannot be held responsible for problems arising from the use of that unsupported version.

Business Hours Support

Your Location	Phone	Fax	E-mail
United States and Canada	800.354.3578	239.263.2883	support@asg.com
Australia	61.2.9460.0411	61.2.9460.0280	support.au@asg.com
England	44.1727.736305	44.1727.812018	support.uk@asg.com
France	33.141.028590	33.141.028589	support.fr@asg.com
Germany	49.89.45716.222	49.89.45716.400	support.de@asg.com
Singapore	65.6332.2922	65.6337.7228	support.sg@asg.com
All other countries:	1.239.435.2200		support@asg.com

Non-Business Hours - Emergency Support

Your Location	Phone	Your Location	Phone
United States and Canada	800.354.3578		
Asia	65.6332.2922	**Japan/Telecom**	0041.800.9932.5536
Australia	0011.800.9932.5536	**Netherlands**	00.800.3354.3578
Denmark	00.800.9932.5536	**New Zealand**	00.800.9932.5536
France	00.800.3354.3578	**Singapore**	001.800.3354.3578
Germany	00.800.3354.3578	**South Korea**	001.800.9932.5536
Hong Kong	001.800.9932.5536	**Sweden/Telia**	009.800.9932.5536
Ireland	00.800.9932.5536	**Switzerland**	00.800.9932.5536
Israel/Bezeq	014.800.9932.5536	**Thailand**	001.800.9932.5536
Japan/IDC	0061.800.9932.5536	**United Kingdom**	00.800.9932.5536
		All other countries	1.239.435.2200

ASG Web Site

Visit http://www.asg.com, ASG's World Wide Web site.

Submit all product and documentation suggestions to ASG's product management team at http://www.asg.com/asp/emailproductsuggestions.asp.

If you do not have access to the web, FAX your suggestions to product management at (239) 263-3692. Please include your name, company, work phone, e-mail ID, and the name of the ASG product you are using. For documentation suggestions include the publication number located on the publication's front cover.

Contents

Preface iii

About this Publication iii

Publication Conventions iv

1 Introduction 1

2 The POST/MAIL Capability 5

Specification of the POST/MAIL Command 6
Examples 8

The POST/MAIL Sequential Output Dataset 9
In Environments Other Than CICS 9
In CICS Environments 10

The CICS Hard Copy Feature 10

3 The Access Call Capability 11

Programming Access Calls 13
User Program Coding 13
Access Call Control Block Settings 15
Installation and Job Control Requirements 16

Parameter Specifications 17

The Access Call Capability Under CICS 27

Programming Access Calls Under CICS 28
User Programming 28
Control Block Settings 31

Installation And Job Control Requirements 32

Storage Area Specifications 33

4 Output Formats 41

Output Records 42

The User Interface Dictionary 44

The User Interface Source Libraries 49

The User Interface Datasets 50

5 The Input User Exit Capability 51

Using the Capability 55
Setting Return Codes 55
Generating a User Exit Diagnostic Message 55
Amending Input Records 55
Rejecting Input Records 56

Tailoring the Capability 57

6 The Access Call Capability Under CICS 59

Programming Access Calls Under CICS 59
User Programming 59
Control Block Settings 62

Installation And Job Control Requirements 64

Storage Area Specifications 64

Index 73

Preface

This *ASG-DataManager User Interface Facility* describes the ASG-DataManager (herein called DataManager) User Interface optional facility. This facility provides DataManager users with three types of interfaces between DataManager and non-ASG-Manager Products software:

- A data dictionary output interface (POST/MAIL capability)
- A dynamic input/output interface (Access Call capability)
- A command input interface (Input User Exit capability)

This manual is intended to be used by those who have a knowledge of DataManager to the extent covered by the *ASG-Manager Products Dictionary/Repository User's Guide*. Users should also have an appreciation of the role of ASG-ControlManager as the interface between DataManager and the ASG-Manager Products end user.

ASG welcomes your comments, as a preferred or prospective customer, on this publication or on the DataManager product.

About this Publication

This publication consists of these chapters:

- Chapter 1, "Introduction," outlines the capabilities of the facility.
- Chapter 2, "The POST/MAIL Capability," describes the POST/MAIL capability of the Facility.
- Chapter 3, "The Access Call Capability," describes the Access Call capability and its use in non-CICS and CICS environments.
- Chapter 4, "Output Formats," describes the formats in which output is generated when the User Interface facility is used, and the data dictionary (User Interface Dictionary) that ASG provides to describe these formats.
- Chapter 5, "The Input User Exit Capability," describes the Input User Exit capability.

Publication Conventions

ASG uses these conventions in technical publications:

Convention	Represents
ALL CAPITALS	Directory, path, file, dataset, member, database, program, command, and parameter names.
Initial Capitals on Each Word	Window, field, field group, check box, button, panel (or screen), option names, and names of keys. A plus sign (+) is inserted for key combinations (e.g., Alt+Tab).
`lowercase italic monospace`	Information that you provide according to your particular situation. For example, you would replace *`filename`* with the actual name of the file.
`Monospace`	Characters you must type exactly as they are shown. Code, JCL, file listings, or command/statement syntax. Also used for denoting brief examples in a paragraph.
Vertical Separator Bar (|) with underline	Options available with the default value underlined (e.g., Y|N).

1 Introduction

These are some of the uses of the User Interface facility:

- To generate source data description statements for a compiler or a database management system for which a DataManager direct interface is not available.
- To control data dictionary updates for adherence to installation standards and naming conventions.
- To interface data dictionary contents to a computerized typesetting program, for publication production. The scope for applications such as this is particularly extensive where the DataManager User Defined Syntax facility is also installed.

Because DataManager is fully integrated with the ASG-Manager Products End User facility, ControlManager, users of the User Interface Facility can also use the facility with certain ControlManager commands. These are the non-interactive commands provided by the ControlManager nucleus (a full description of these commands is given in the *ASG-ControlManager User's Guide*). Thus users can use the POST/MAIL capability to save the output of these commands, can execute these commands from user-written programs by invoking the Access Call capability, and can examine individual ControlManager input records by using the Input User Exit capability. In the remainder of this publication whenever DataManager commands, input records or output are mentioned, then the mention of ControlManager non-interactive commands, input records or output is also implied.

The User Interface facility provides three capabilities for DataManager users. Each of these capabilities provides a certain type of interface between DataManager and user-written programs:

- The output interface provided by the POST/MAIL capability. This capability enables users to use the POST/MAIL command, which when prefixed to another DataManager command, directs the output of the latter to a sequential dataset on disk or tape, for subsequent processing by user-written programs. This output can be additional to or an alternative to the normal printed or displayed output that is generated by the prefixed command.

- A dynamic input/output interface, which is invoked by calls from user-written programs. This capability is termed the Access Call capability. When invoked through this interface, DataManager is not run independently, but is attached dynamically as a result of calls from the user-written program. Any DataManager commands can be passed to DataManager by Access Calls. Output from the execution of these commands is placed directly by DataManager into a buffer which is supplied by the user program and is specified as a parameter of the CALL statement that invokes an Access Call.
- A command interface (Input User Exit capability) which consists of a user exit in the DataManager program. After each input record is read, control is passed via the user exit to a routine where the input record can be inspected and, if desired, amended by user-written code within the routine. User-written code can also set a return code to DataManager to indicate whether the current input record is to be accepted or is to be rejected with an error message.

The Access Call and POST/MAIL capabilities are not mutually exclusive. For example, it is possible to execute POST/MAIL commands through Access Calls.

Users can make use of the POST/MAIL capability when they are using DataManager in any of these environments:

- OS and DOS batch environments
- CICS, CMS, IMS/DC, ROSCOE, TSO, and TSO/ISPF

Full details of the POST/MAIL capability are given in Chapter 2, "The POST/MAIL Capability," on page 5.

The Access Call capability can be used to enter ASG-ControlManager (and then DataManager) from any of these environments:

- OS and DOS batch environments.
- CICS, CMS, TSO, and TSO/ISPF. In addition, users can use the Access Call capability from ROSCOE and IMS/DC environments if they can ensure that all the requirements of these environments are satisfied when an Access Call is made.

Full details of the Access Call capability are given in Chapter 3, "The Access Call Capability," on page 11. When operating under CICS, the programming and parameter requirements are different from those of other environments, and the requirements for the use of Access Calls under CICS are given separately in Chapter 6, "The Access Call Capability Under CICS," on page 59.

The format of output from POST/MAIL commands and from Access Calls are similar, and are described in Chapter 4, "Output Formats," on page 41.

Users can make use of the Input User Exit capability in any environment in which DataManager is running.

Full details of the Input User Exit capability are given in Chapter 5, "The Input User Exit Capability," on page 51.

Throughout this publication, the terms *user-written program* and *user program* are used to refer to any non-Manager Products program code that is involved in the use of the User Interface facility.

2 The POST/MAIL Capability

The POST/MAIL capability provides a means whereby the user can create a sequential dataset containing the output generated by one or more DataManager commands. This output is then available for subsequent processing by user-written programs. Thus the capability facilitates the use of information from a data dictionary for purposes that are not directly supported by DataManager.

The POST/MAIL capability is invoked by using the POST/MAIL prefixing command. (The two keywords POST and MAIL are synonymous.) A POST/MAIL command may prefix any DataManager command except a DICTIONARY command, a CLOSE command, a PERFORM command, or a Controller's CREATE command or RELOAD command. (It is, however, possible to POST the output of PERFORMed commands: see "Specification of the POST/MAIL Command" on page 6.) The action of the POST/MAIL command is to place the output of the command that it prefixes into the sequential dataset.

A variant of the POST or MAIL command enables the user to specify the dataset that is to receive the output, so that a number of different POST/MAIL output datasets can be created in the same run. This dataset is referred to as the *POST/MAIL Sequential Output Dataset*.

A user must open a data dictionary before he or she issues any POST/MAIL command.

Figure 1 on page 7 outlines the action of the POST/MAIL capability.

Specification of the POST/MAIL Command

The POST/MAIL command writes the output generated by a prefixed DataManager command to the primary output device and/or the current POST/MAIL Sequential Output Dataset, or you can specify a new POST/MAIL Sequential Output Dataset.

This is the format of the POST/MAIL command:

```
{ POST }  { [AND] command                                  }   { ; }
{ MAIL }  { [NORMAL] ONTO { filename      }                }   { . }
          {               { destinationID }                }
```

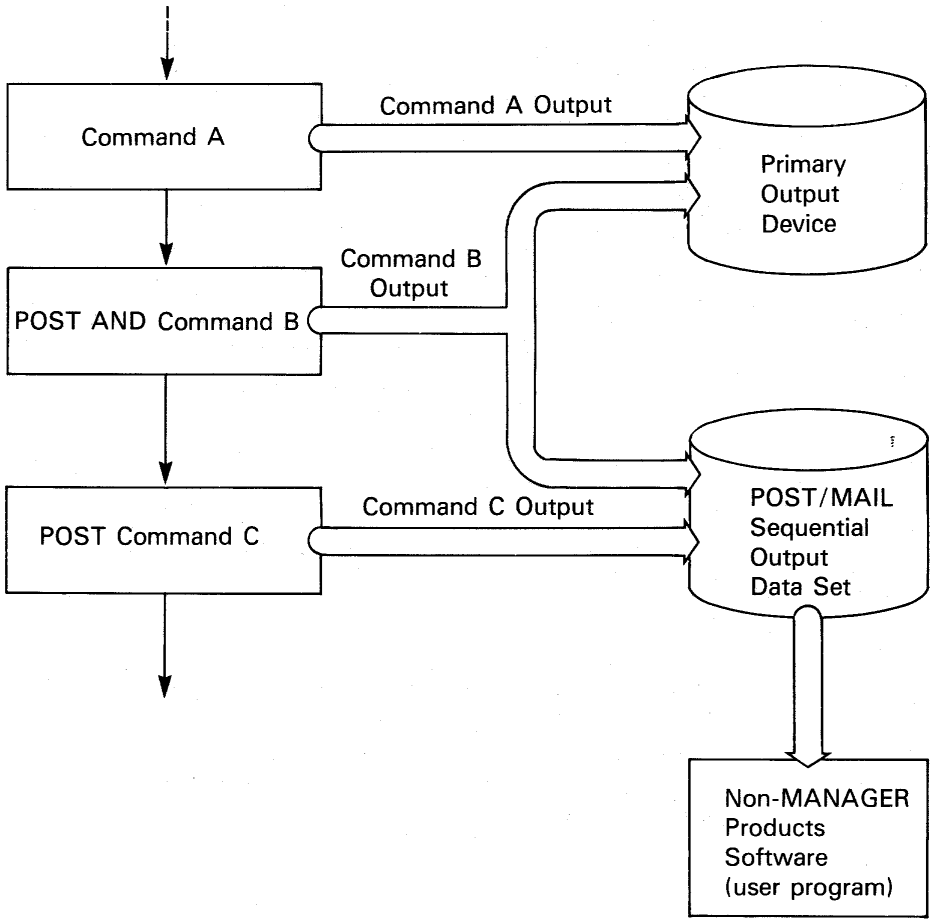
where:

command is any DataManager command (excluding its terminator) except DICTIONARY, CLOSE, PERFORM, CREATE, or RELOAD

filename is the logical name (DTF name under DOS or DD name under OS) of the dataset. *filename* can have a maximum of seven characters under DOS or a maximum of eight characters under OS.

destinationID is the logical name of the Extrapartition dataset that is to receive the output from subsequent POST/MAIL commands when operating under CICS. The name must have been specified in the CICS Destination Control Table as a four-character Destination Identification; *destinationID* is truncated from the right to four characters if it is declared as more than four characters in the POST/MAIL command.

Figure 1 • The POST/MAIL Capability



Remarks

1 The command identifiers POST and MAIL are alternatives having the same meaning.

2 If a POST/MAIL command prefixes a command, then the prefixed command is first executed and its output (if any) is directed to the datasets specified in Remarks 3 and 4.

3 If a POST/MAIL command includes the AND keyword (and therefore prefixes a command) then the output generated by the prefixed command is written both to the primary output device (MPOUT or SYSLST) and the POST/MAIL Sequential Output Dataset.

4. If a POST/MAIL command prefixes a command, but does not include the AND keyword, then the output generated by the prefixed command is directed only to the current POST/MAIL Sequential Output Dataset.

5. A POST [AND] PERFORM command will not give the required results due to the way in which the PERFORM command operates (see the command's specification in the *ASG-Manager Products Dictionary/Repository User's Guide*). This command:

   ```
   PERFORM "POST [AND] command;";
   ```

   should be used instead to POST the output resulting from a PERFORMed command.

6. Except when operating under CICS, if ONTO *filename* follows the POST or MAIL command identifier, then:

   - If there is a POST/MAIL Sequential Output Dataset already open, then it is closed
   - When the next command prefixed by POST or MAIL is executed, an attempt is made to open a new dataset called *filename*. If this attempt is successful, then *filename* serves as the Sequential Output Dataset until a further POST ONTO command is issued.

7. When operating under CICS, if ONTO *destinationID* follows the POST or MAIL command identifier, then the output from subsequent prefixing POST/MAIL commands is directed to the destination specified by *destinationID*.

8. If the keyword NORMAL precedes the ONTO clause, then until a further POST ONTO or MAIL ONTO command is accepted, DataManager line printer output images are output by subsequent POST or MAIL commands, instead of POST/MAIL fixed-format records (as discussed in "Output Records" on page 42).

9. POST/MAIL Sequential Output Datasets are opened and closed as described in "The POST/MAIL Sequential Output Dataset" on page 9.

Examples

```
POST AND GLOSSARY FILES GIVING NOTES;
```

The output from the GLOSSARY command is written to both the primary output device and the current POST/MAIL Sequential Output Dataset.

```
POST BULK PRINT CHANGED MODULES;
```

The output from the BULK PRINT command is only written to the current POST/MAIL Sequential Output Dataset.

```
POST ONTO MYPOST.
```

A dataset that is identified in the job control by the logical file name MYPOST is opened as the POST/MAIL Sequential Output Dataset. If there is any dataset already open as the POST/MAIL Sequential Output Dataset, then it is closed.

The POST/MAIL Sequential Output Dataset

In Environments Other Than CICS

A POST/MAIL Sequential Output Dataset is opened when the first POST/MAIL command in a run is accepted. If this command includes the ONTO keyword, then the dataset specified in the command is opened. If this command does not include the ONTO keyword, then a dataset called MPRPOST is immediately opened as the POST/MAIL Sequential Output Dataset and the POST/MAIL command is then executed as described above.

Output is written to the POST/MAIL Sequential Output Dataset from each POST/MAIL command until one of the following events occurs:

- The run ends
- A DICTIONARY command or a CLOSE command is processed
- A POST ONTO command is processed

The dataset is then closed. Subsequent action depends on which of these events occurred. If a POST ONTO command was processed, then an attempt is made to open the dataset named in the POST ONTO command (as described in "Specification of the POST/MAIL Command" on page 6. See Remark 6). If a DICTIONARY or a CLOSE command was processed, then no further action is taken immediately, but the next POST command processed causes a dataset to be opened as a POST/MAIL Sequential Output Dataset, as described at the start of this section.

The POST/MAIL Sequential Output Dataset has a variable length record format and is blocked.

Care should be taken to ensure that the same dataset is not opened as the POST/MAIL Sequential Output Dataset more than once in each session. This may happen if the same data dictionary is opened more than once and POST/MAIL commands are issued each time it is open. If this is done, the POST/MAIL Sequential Output Dataset will only contain the POST/MAIL output from the last period in which the dictionary was open.

For installation requirements and sample job control, refer to one the manual that is appropriate to the operational environment:

- *ASG-Manager Products Installation in OS Environments*
- *ASG-Manager Products Installation in DOS Environments*
- *ASG-Manager Products Installation in CMS Environments*

In CICS Environments

When running DataManager under CICS, each user must issue a POST ONTO command before issuing his or her first prefixing POST/MAIL command. A separate POST/MAIL destination (dataset) can thus be established for each user. The `destinationID` specified in a POST ONTO command remains effective until a further POST ONTO command is issued or until the end of the session. If the `destinationID` specified in the command is longer than 4 characters, it is truncated from the right to 4 characters by DataManager; if it is otherwise invalid, or is omitted, the command is rejected, and no prefixing POST/MAIL commands are accepted from that user until a valid POST ONTO command is received.

Under CICS, users must include explicit statements to open and close any POST/MAIL destination dataset. A POST/MAIL destination dataset is defined as variable blocked with a maximum record size of 600 bytes. Under DOS, for compatibility with batch output, the blocksize should be 3156 bytes.

For installation requirements and sample job control, refer to the relevant publication:

- *ASG-Manager Products Installation in OS Environments*
- *ASG-Manager Products Installation in DOS Environments*

The CICS Hard Copy Feature

When it is installed in a CICS environment, the POST/MAIL capability also provides a Hard Copy feature for the DataManager user. This feature enables users to switch command output from the primary output device to a specified local printer by using the POST/MAIL command. This Feature is fully described in the CICS section of the Manager Products installation manuals.

3 The Access Call Capability

The Access Call capability allows user programs, written in a variety of programming languages, to call ControlManager, and subsequently DataManager. This facilitates a dynamic use of a data dictionary, because the results of Access Calls can control the user program's processing.

The Access Call capability enables a user:

- To access data dictionaries from a user program, without needing to know DataManager's internal structure or processing techniques
- To use the information stored in a data dictionary to drive, or to act as a reference file for, a user program

An Access Call to DataManager is made using a standard programming language CALL statement. This typically specifies what is to be called together with certain parameters: the main storage areas that contain data for, or data returned by, the called program. The data for the called program, in the case of the DataManager Access Call, includes a sequence of one or more DataManager commands. The called program is an ASG-supplied interface routine that is link edited with the user program. The interface routine calls ControlManager which in turn calls DataManager to process the sequence of commands. The output generated by these commands is returned to the user program a record at a time by repeated further Access Calls.

There is no limitation on the types of DataManager commands that users can execute by means of the Access Call capability: System Administrator's and Controller's commands and the ability to read and update a data dictionary are all allowed. However, commands used in Access Calls are subject to their normal security rules and to the security levels established by Controllers or the Systems Administrator.

Thus the Access Call capability provides a high level interface that requires no knowledge of the internal structure of DataManager. All that is needed is an understanding of DataManager's standard data definition language, commands, and the related outputs.

The use of Access Call in CICS environments differs in detail from its use in other environments. The use of Access Call under CICS is therefore documented separately in Chapter 6, "The Access Call Capability Under CICS," on page 59.

These figures illustrate how the Access Call capability operates in non-CICS environments. Figure 2 shows the way that control passes between the user program, the interface routine, ControlManager, and DataManager. Figure 3 shows the way in which DataManager input records, and the corresponding generated output records, are routed between the user program, ControlManager and DataManager.

Figure 2 • The Access Call Capability (Control Flow)

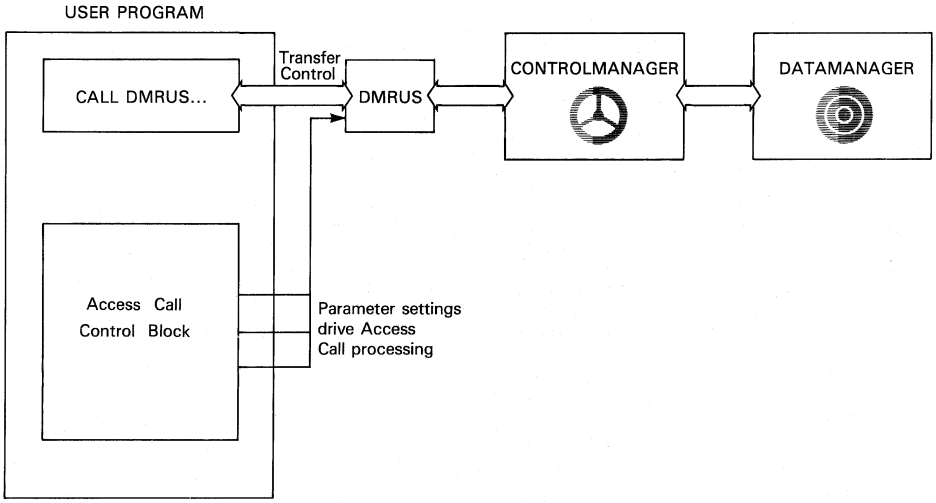
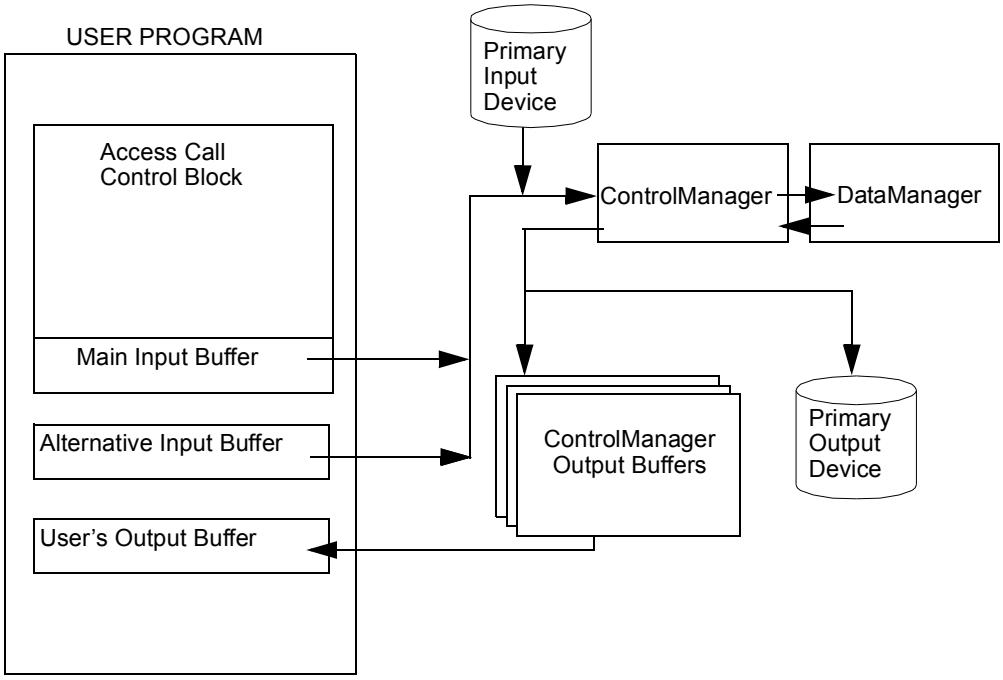


Figure 3 • The Access Call Capability (Data Flows)



Programming Access Calls

User Program Coding

An Access Call is a standard programming language CALL. It can therefore be used in programs written in any language that supports such a CALL; for example, COBOL, PL/I, Assembler, or FORTRAN. The syntax of a CALL varies from language to language, and in some cases, between different implementations of a language. In this manual, conceptual syntax is used. It is assumed that the user has a sufficient knowledge of the relevant programming language to be able to code the required CALLs from these conceptual specifications.

Conceptually, the Access Call statement is of this form:

```
CALL entry-point parameter-list
```

where:

> *entry-point* specifies an entry point name in the Access Call Interface Routine that is link edited with the user program. This entry point name is DMRUS, and is applicable for COBOL, FORTRAN, Assembler or PL/I Optimizer user programs. PL/I F level user programs are not supported. PL/I Optimizer programs that use the Access Call capability must include this statement:
>
> ```
> DCL DMRUS ENTRY OPTIONS (COBOL,NOMAP);
> ```
>
> *parameter-list* comprises either two or three parameter names. These names are the symbolic names of main storage areas that are to contain data for the Interface Routine or that are to receive data from the Interface Routine.

The first parameter in *parameter-list* names a control block (called the Access Call Control Block). This control block contains a number of control fields (defined in "Parameter Specifications" on page 17) together with an input buffer. The input buffer can be filled (by the user program at run time) with DataManager input records. The user may include up to 32767 input records at one time. When the CALL is executed, all the input in the buffer is processed as a single *request* to DataManager. Alternatively, particular Access Call Control Block fields can be set to obtain the input command stream from an input dataset (MPIN or SYSIPT) or from another main storage area named by the optional third parameter of the CALL list.

The second parameter in *parameter-list* names the main storage area that is to receive each output record returned to the user program.

The optional third parameter in *parameter-list* is the name of a main storage area from which input can be obtained (if the appropriate field of the Access Call Control Block is set) instead of from the input buffer that follows the Access Call Control Block.

Any symbolic names allowable in the programming language in which the user program is coded can be used for these parameters. In this manual, these names are used:

- DCONTROL for the first parameter
- DOUTPUT for the second parameter

No standard name is applied in this manual to the optional third parameter.

In addition to the description of the structure of the Access Call Control Block in this chapter, the structure of the Access Call Control Block is also described in a machine-readable form. This second description takes the form of a series of interrelated data dictionary member definitions. These dictionary members form part of the User Interface Dictionary which is supplied to users on the installation tape. Several programming language copies of these definitions are also included on the installation tape, in order to allow inclusion of the appropriate copy into user programs. This second description of the Access Call Control Block, and a general description of the information provided by the contents of the User Interface Dictionary, is outlined in Chapter 4, "Output Formats," on page 41.

A definition of the last field in the DCONTROL data structure, DINPUT, is not given, because its length depends on user requirements. Users must therefore define DINPUT themselves.

Specifications of the individual fields of the two named parameters DCONTROL and DOUTPUT, are given in "Parameter Specifications" on page 17. The field names there used are those used for the corresponding data definitions in the User Interface Dictionary. Other names can be substituted in user programs if required.

The first command processed by the first Access Call of a session should be a LOGON or a DICTIONARY command. If it is a LOGON command, then the user is logged on to ControlManager under the logon identifier specified. If it is not a LOGON command, then the user is logged on to ControlManager by the Autolog process and the specified data dictionary is opened.

The last command processed by the final Access Call of a session should be a LOGOFF command. This command closes any open data dictionary and datasets, and logs the user off from ControlManager. One of the fields of the Access Call Control Block (DRETURN) has a value of 3 whenever a LOGOFF command has been processed.

For upwards compatibility, users may continue to use an ENDDMR command instead of LOGOFF as the final command. DRETURN is also set to 3 after processing of an ENDDMR command. All DataManager command syntax rules apply equally to commands executed using the Access Call capability. Terminators must be included, both for commands and for the data definitions or amendments associated with any INSERT, ADD, ALTER, MODIFY or REPLACE commands. The whole of any such command and its associated data definition or amendment must be processed in a single Access Call.

Access Call Control Block Settings

The details of initializing or setting Control fields to control the various options available within Access Call are given in the specifications of the parameters in "Parameter Specifications" on page 17. If a field is defined as initialized by the user program, its value must be set by the user program for the first CALL to operate on a particular dictionary, and must not be altered by the user program until that dictionary is closed. If a field is defined as set by the user program, it can be reset for later CALLs. User programs must not modify the contents of any field of DCONTROL unless the specification of the field states that the contents are to be set or initialized.

When a request is processed by an Access Call, the DataManager output is placed first in one or more ControlManager buffers. From these ControlManager buffers, the Access Call returns one output record at a time to the user program buffer DOUTPUT. An output record can be a line output by a DataManager command, or a message as defined in the Messages manual or the Restricted Messages manual. A control field, DRETURN, informs the user program whether the content of DOUTPUT is a line or a message, or whether nothing is returned. In the latter case, tests on another control field, DRETURN2, can ascertain the reason. When control is returned to the user program after an Access Call, DRETURN should be the first field examined, followed if necessary by DRETURN2. If DRETURN2 shows a return code of 3, users should attempt to run the failed command in a non-Access Call batch environment in order to ascertain the cause of the failure.

It was stated earlier that a request could be a single DataManager command or a stream of DataManager commands, held in the input buffer (DINPUT). As the output from a request is placed in the user program buffer DOUTPUT one record at a time, it is necessary to have a mechanism by which the second and subsequent output lines or messages can be transferred to this buffer for processing. This mechanism is provided by the control field DFUNC. When servicing any request, the value of DFUNC is first set to indicate that the Access Call is supplying a request from DINPUT. The CALL is then made; if successful, it leaves the first output record in DOUTPUT. After processing this record, the user program alters the value in DFUNC to indicate that the next output record is to be placed in DOUTPUT and repeats the CALL. This process repeats until all output records resulting from the request have been processed. Tests on another control field, DOUTWAIT, can determine how many output records are still waiting in the ControlManager buffer for transfer to DOUTPUT (but DOUTWAIT should not be tested alone for end of output, as this could leave a line or message unprocessed in DOUTPUT; DRETURN should be tested first, as stated in the previous paragraph).

The user can select, by setting a value in DOPTION5, whether the output records placed from the ControlManager buffers into DOUTPUT are to comprise all lines and messages output, or only the lines, or only the messages. The placing of lines or messages into DOUTPUT does not destroy their records in the ControlManager output buffers; so the user can elect to process all messages first, and then proceed to process the output lines; or to process the output lines first, followed by the messages; or to process them together in the order in which they were output; or can switch between lines and messages at any time by resetting the values in DOPTION5 and DFUNC between CALLs.

The values set in other control fields can similarly be varied between CALLs to control the processing; the fields for which this applies have the remark `Must be set by user program` in their specification in "Parameter Specifications" on page 17. The user should note, however, that many control fields are not examined on every Access Call; whether they are examined is dependent upon the value in DFUNC. Where this applies, this fact is also stated in the specification in "Parameter Specifications" on page 17.

If it is requested to have more than one data dictionary open at the same time, then each must have its own DCONTROL parameter. It is possible to share DOUTPUT, but this is not recommended. If two or more data dictionaries are to be accessed but only one is to be open at any time, then the same parameter areas may be used to access each data dictionary.

When testing an Access Call program with input from DINPUT or from a parameter area (DOPTION2=0 or 3), it is suggested that the user should set DOPTION1 to 1 and DOPTION2 to 3. This will display all DataManager commands, output lines, and messages on the normal ControlManager primary output device.

Installation and Job Control Requirements

For the linkage editing of user programs that utilize Access Calls and for the job control requirements for the running of such programs, refer to the appropriate publication:

- *ASG-Manager Products Installation in OS Environments*
- *ASG-Manager Products Installation in DOS Environments*
- *ASG-Manager Products Installation in CMS Environments*

Parameter Specifications

The structure of the parameters used in Access Calls are defined in the following table. The field names given in the table are those used in the User Interface Dictionary. The User Interface Dictionary is described in Chapter 4, "Output Formats," on page 41.

Access Call Parameter Fields

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
Control parameter group, containing:	DCONTROL	0	GROUP	—	—	This field's name is the first parameter in an Access Call. Length in bytes is 112 plus the length of DINPUT.
Communication field	DCOMMUNE	0	Character	64	—	Reserved for ControlManager use. Must be initialized by user program to binary zero/'LOW VALUES' before the first Access Call.

Access Call Parameter Fields

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
ControlManager program or server name	DDMR	64	Character	8	Non-zero	Must be initialized by the user program to specify whether non-server (standard) mode or server (ASG-Manager Products Server Facility) mode execution is required. For standard execution specify these values: • MPR00 • CM00 • CM01 • CM02 (CMS only) • CM03 (CMS only) • DM00 • DM01 For MPSF execution specify the name of the required server.
ControlManager output buffer size	DBUFFLEN	72	Binary	4	- Zero	No longer used. Any value specified is ignored.

Access Call Parameter Fields

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
Input buffer size/Input line count	DINPLEN	76	Binary	2	Non-zero (max. 32767)	Examined only when DFUNC=1. Specifies the size of the user program input buffer used when DOPTION2=0, 3, 4, or 5, as specified by DOPTION8, as follows. If DOPTION3=0, then DINPLEN gives the buffer size in bytes and this value should be a multiple of the value of DINPLREC. If DOPTION3=1, then DINPLEN gives the number of input records in the buffer.
Input line size	DINPLREC	78	Binary	2	Non-zero (max. 255)	When DOPTION2=0, 3, 4, or 5, must be set by the user program with the size in bytes of the input lines. The maximum size is 255. If DOPTION3=0, the size must not be greater than the value in DINPLEN and should divide exactly into the value in DINPLEN. DINPLREC is examined only when DFUNC=1.
Output record maximum size	DOUTLEN	80	Binary	2	Non-zero (max. 600)	Must be set by user program with the maximum size of output record to be held in DOUTPUT. If the value specified is less than the maximum possible output record size, then output records that exceed the specified maximum size will be truncated from the right. If truncation of the record in DOUTPUT occurs, the original length of the output line or message is given in DOUTLREC. DOUTLEN is only examined when DFUNC=1 or 2 or 3.

Access Call Parameter Fields

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
Output record actual size	DOUTLREC	82	Binary	2		Returned by ControlManager. Gives the original length in bytes of the output record placed in DOUTPUT. This length will be greater than the value in DOUTLEN if truncation of the output record has occurred.
Function required	DFUNC	84	Numeric-character	1		Must be set by user program to indicate the function required by the current Access Call, thus:
					'1'	• This call is supplying a new request
					'2'	• The next output line or message is to be placed in DOUTPUT
					'3'	• ControlManager is to retrieve the first output line or message and place it in DOUTPUT.
Return code, contents of DOUTPUT	DRETURN	85	Numeric-character	1		Returned by ControlManager. Indicates what has been placed in DOUTPUT, thus:
					'0'	• Content of DOUTPUT is a line.
					'1'	• Content of DOUTPUT is a diagnostic message.
					'2'	• Nothing has been placed in DOUTPUT.
					'3'	• Nothing has been placed in DOUTPUT and ControlManager has terminated.
Return code, Access Call work file	DRETURN1	86	Numeric-character	1	'1'	No longer used. For upwards compatibility is always set to value '1'.

Access Call Parameter Fields

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
Return code, input condition	DRETURN2	87	Numeric-character	1		Returned by ControlManager. Indicates the result of input processing, thus:
					'0'	• Input was processed.
					'1'	• End of data on the input dataset (MPIN or SYSIPT) was detected.
					'2'	• The input dataset (MPIN or SYSIPT) could not be opened, as its file definition was omitted from the job control.
					'3'	• A serious error condition occurred, which caused processing of the request to be terminated.
					'4'	• User of the BREAK key interrupted the run.
					'5'	• Unable to connect to server.
					'6'	• Client time-out occurred.
					'7'	• Server task has abended.
					'8'	• Unexpected server shutdown has occurred.
						Note: Return codes 5 through 8 apply only when executing with Manager Products Server Facility.

Access Call Parameter Fields

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
						Return codes 1 and 2 apply only if DOPTION=1, 2, 4, or 5.
Output message severity code	DSEVRITY	88	Character	1		Returned by ControlManager. Indicates the highest severity code of any diagnostic message in the ControlManager output buffer, thus:
					Space	• No Messages Output
					'I'	• One or more informatory messages output
					'W'	• One or more warning messages output
					'E'	• One or more error messages output
					'S'	• One or more serious messages output
					'C'	• A critical message output.
Print option	DOPTION1	89	Numeric-character	1		Must be set by user program to indicate print option, thus:
					'0'	• Neither output nor input to be printed
					'1'	• Output to be printed on output dataset (MPOUT or SYSLST). Input also printed if DOPTION2=2 or 3.
						Examined only if DFUNC=1.

Access Call Parameter Fields

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
Input option	DOPTION2	90	Numeric-character	1		Examined only if DFUNC=1. Must be set by user program to indicate input option, thus:
					'0'	• Read input from DINPUT or from parameter area, depending on DOPTION8.
					'1'	• Read input from input dataset (MPIN or SYSIPT).
					'2'	• Read input from input dataset (MPIN or SYSIPT), and print it if DOPTION1=1.
					'3'	• Read input from DINPUT or from parameter area, depending on DOPTION8, and print it if DOPTION1=1.
					'4'	• Read input from DINPUT or from parameter area, depending on DOPTION8, followed by input from input dataset (MPIN or SYSIPT). On conclusion of input, DOPTION2 is reset to 1.
					'5'	• Read input from DINPUT or from parameter area, depending on DOPTION8, followed by input from input dataset (MPIN or SYSIPT). Print the input if DOPTION1=1. On conclusion of input, DOPTION2 is reset to 2.

Access Call Parameter Fields

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
						DOPTION2=2 or 3 is normally used with DOPTION4=1, but can be used with other values for DOPTION4.
Input size option	DOPTION3	91	Numeric-character	1		Specifies use of DINPLEN:
					'0'	• DINPLEN holds the size in bytes of the input buffer used.
					'1'	• DINPLEN holds number of lines of input in the input buffer used.
Input processing option	DOPTION4	92	Numeric-character	1		Must be set by user program if DOPTION2=1 or 2, to indicate multiple or single command processing from the input dataset:
					'0'	• Return to user program after all input lines are processed
					'1'	• Return to user program after each command is processed
						Examined only if DFUNC=1.
Output option	DOPTION5	93	Numeric-character	1		Must be set by user program to indicate output option:
					'0'	• Only output lines to be placed in DOUTPUT
					'1'	• Only messages to be placed in DOUTPUT
					'2'	• Output lines and messages to be placed in DOUTPUT in the order in which they were generated

Access Call Parameter Fields

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
						Examined only if DFUNC=1 or 3.
Output format	DOPTION6	94	Numeric-character	1		Must be set by user program to indicate the output format required:
					'0'	• Lines in DOUTPUT to be in same format as POST/MAIL output
					'1'	• Lines in DOUTPUT to be in normal output print format
						Examined only if DFUNC=1.
Access Call Work File usage	DOPTION7	95	Numeric-character	1	-	No longer used. Any value specified is ignored.
Input buffer selection	DOPTION8	96	Numeric-character	1		Must be set by user program if DOPTION2=0, 3, 4, or 5, to indicate the location of the input, thus:
					'0'	• Input is in DINPUT
					'1'	• Input is in the area defined by the third parameter of the current Access Call
						Examined only if DFUNC=1.
Execution mode	DOPTION9	97	Numeric-character	1		Must be initialized by user program to indicate the execution mode, thus:
					'0'	• Execution is batch.
					'1'	• Execution is interactive or online. If DataManager is running under TSO, BREAK key or PA1 key attention interruption is enabled.

Access Call Parameter Fields

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
					'2'	• Execution is interactive or online. If DataManager is running under TSO, BREAK key or PA1 key attention interruption is not enabled.
Count of output records queued	DOUTWAIT	98	Packed-decimal	3	five digits	Set by ControlManager. Indicates the number of output records still waiting in the ControlManager output buffer to be placed in DOUTPUT.
Number of output lines generated	DOUTTOTL	101	Packed-decimal	3	five digits	Returned by ControlManager when DFUNC=1. Indicates the number of output lines generated by the Access Call request.
Number of message lines generated	DOUTTOTM	104	Packed-decimal	3	five digits	Returned by ControlManager when DFUNC=1. Indicates the number of message lines generated by the Access Call request.
Command number	DINPNO	107	Packed-decimal	2	three digits	Returned by ControlManager to indicate the number of the command to which the output record relates. The commands are numbered from 1 in increments of 1. The command number is reset to 1 when a CALL is made with DFUNC=1.
Reserved	DRESERVE	109	Character	3		Reserved for future use.
Input buffer	DINPUT	112	Character	As defined by DINPLEN		When DFUNC=1 and DOPTION8=0, must be set by user program with DataManager input lines (each of the length specified in DINPLREC) to constitute one Access Call.

Access Call Parameter Fields

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
Output buffer: group, containing:	DOUTPUT	0	GROUP	As defined by DOUTLEN		This field name is the second parameter in each Access Call. This field is used by ControlManager to return output records resulting from the user's CALL. One record is returned by each CALL. A record can be an output line or a message. If the output line or message is shorter than the length of DOUTPUT (as specified in DOUTLEN), DOUTPUT is space-filled to the right. Chapter 4, "Output Formats," on page 41 describes how to ascertain the formats of output records.
Character of output	DOUTCHAR	0	Character	1		

The Access Call Capability Under CICS

Figure 4 on page 28 and Figure 5 on page 29 illustrate how the Access Call capability operates in CICS environments. Figure 4 on page 28 shows the way that control passes between a user program, ControlManager and DataManager. Figure 5 on page 29 shows the way in which DataManager input records, and the corresponding generated output records, are routed between a user program, ControlManager and DataManager.

Programming Access Calls Under CICS

User Programming

Under CICS, an Access Call is carried out by means of a CICS link to program DMGRDMRU. The Access Call can be coded as a macro level call of this form:

```
DFHPC TYPE=LINK.PROGRAM=DMGRDNRU
```

or as a command level call of this form:

```
EXEC CICS LINK PROGRAM('DMGRDMRU')
```

DMGRDMRU corresponds to the interface routine DMRUS which is used in non-CICS environments. When DataManager has completed processing the Access Call, control is returned to the user program.

At any stage of its execution, the user program may re-enter CICS between Access Calls (for example, to read input from the terminal). All the ControlManager and DataManager-related storage areas that need to be preserved for subsequent Access Calls are written to temporary storage records and retrieved as necessary.

ControlManager uses three storage areas, which must be defined to follow the Task Control Area (TCA) as the first fields in the user's Transaction Work Area (TWA). Any other user defined fields in the Transaction Work Area must follow these first three areas.

Figure 4 • The Access Call Capability under CICS (Control Flow)

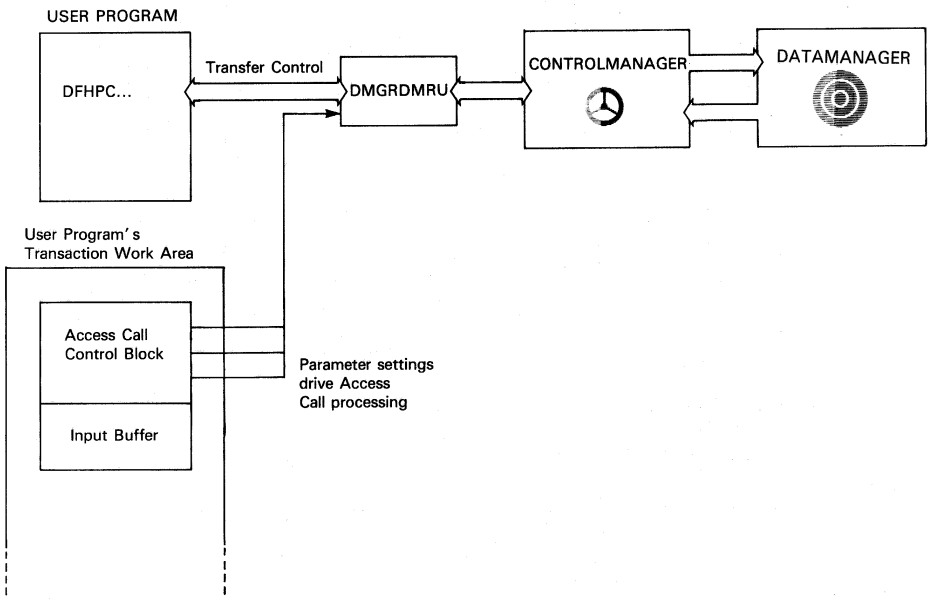
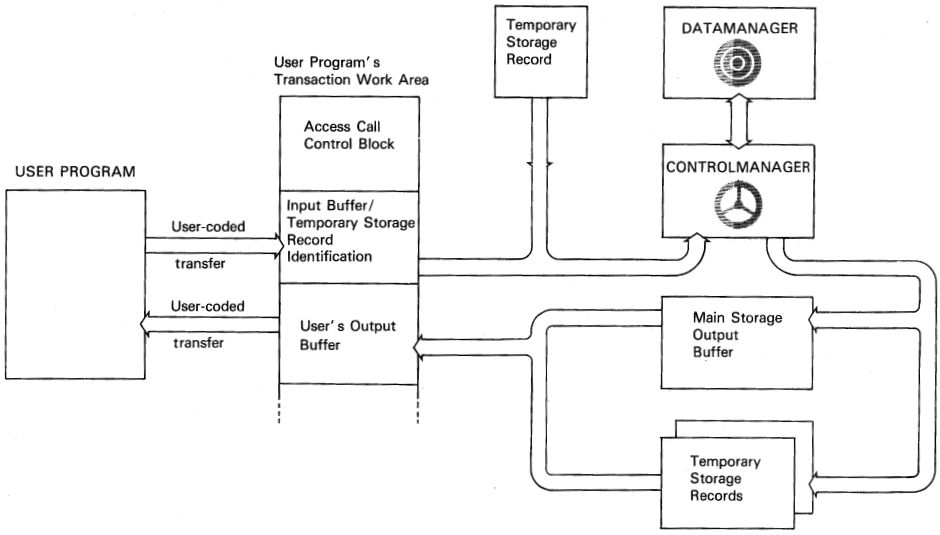


Figure 5 • The Access Call Capability under CICS (Data Flow)



The first of these storage areas is a control block (Access Call Control Block). This block contains a number of control fields (as defined in the table in "Storage Area Specifications" on page 33). The second area is an input buffer. The input buffer can be filled by the user program at run time with DataManager input lines. When the Access Call is performed, all the input in the buffer is processed as a single request to DataManager. A request can be a simple request to process one command only, or a complex request to process a stream of commands.

As an alternative to having input commands held in the above-mentioned buffer in the Transaction Work Area, users can specify that DataManager input is to be read from a temporary storage record. In this case, the input buffer in the Transaction Work Area contains only the eight-character identification of the temporary storage record that contains the input. The User Interface Facility decides which of these two areas contains the current input by comparing the first four characters of the Transaction Work Area input buffer with the four-character identification of the terminal that is associated with the user's transaction; if these are the same, then input is assumed to be held in a temporary storage record whose identification is given by the first eight characters in the Transaction Work Area input buffer, otherwise the input buffer itself is assumed to hold the input. If a temporary storage record is used to hold input, then the storage record is released when the input has been read.

The third of these storage areas receives each output record returned to the user program.

Any symbolic names allowed in the programming language in which the user program is coded can be used to name these three user program storage areas. In this publication, these names are used:

- DCONTROL for the Access Call Control Block
- DINPUT for the input buffer
- DOUTPUT for the output buffer.

In addition to the description of the structure of the Access Call Control Block in this chapter, the structure of the Access Call Control Block is also described in a machine-readable form. This second description takes the form of a series of interrelated data dictionary member definitions. These dictionary members form part of the User Interface Dictionary which is supplied to User Interface facility users on their installation tape. Several programming language copies of these definitions are also included on installation tapes, in order to allow the inclusion of the appropriate copy into user programs. This second description of the Access Call Control Block, and a general description of the information provided by the contents of the User Interface Dictionary, is given in Chapter 4, "Output Formats," on page 41.

A definition of the last field in the DCONTROL data structure, DINPUT is not given, because its length depends on user requirements. Users must therefore define DINPUT themselves.

Specifications of the individual fields of the three named areas, DCONTROL, DINPUT, and DOUTPUT are given in "Storage Area Specifications" on page 33. The field names there used are those used for the corresponding data definitions in the User Interface Dictionary. Other names can be substituted in user programs if required.

The first command processed by the first Access Call of a session should be a LOGON or a DICTIONARY command. If it is a LOGON command, then the user is logged on to ControlManager under the logon identifier specified. If it is not a LOGON command, then the user is logged on to ControlManager by the Autolog process and the specified data dictionary is opened.

The last command processed by the final Access Call of a session should be a LOGOFF command. This command closes any open data dictionary and datasets, and logs the user off from ControlManager. One of the fields of the Access Call Control Block (DRETURN) has a value of three whenever a LOGOFF command has been processed.

For upwards compatibility, users may continue to use an ENDDMR command instead of LOGOFF as the final command. DRETURN is also set to three after processing of an ENDDMR command.

All DataManager command syntax rules apply equally to commands issued via Access Call. Terminators must be included, both for commands and for the data definitions or amendments associated with any INSERT, ADD, ALTER, MODIFY, or REPLACE commands. The whole of any such command and its associated data definition or amendment must be contained in a single Access Call.

Control Block Settings

The details of initializing or setting fields to control the various options available within Access Call are given in the specifications of the storage areas in "Storage Area Specifications" on page 33. If a field is defined as initialized by the user program, its value must be set by the user program for the first call to operate on a particular dictionary, and must not be altered by the user program until that dictionary is closed. If a field is defined as set by the user program, it can be reset for later calls. User programs must not modify the contents of any field of DCONTROL unless the specification of the field states that the contents are to be set or initialized.

When a request is processed by an Access Call, the DataManager output is placed first in a ControlManager buffer. The size of this buffer is determined by the control field DBUFFLEN. From this ControlManager buffer, the Access Call returns one output record at a time to the user program buffer DOUTPUT. An output record can be a line output by a DataManager command, or a message as defined in the *ASG-Manager Products Messages* manual or the Restricted Messages manual. A control field, DRETURN, informs the user program whether the content of DOUTPUT is a line or a message, or whether nothing is returned. In the latter case, tests on the DRETURN2 control fields can ascertain the reason. When control is returned to the user program after an Access Call, DRETURN should be the first field examined, followed if necessary by DRETURN2. If DRETURN2 shows a return code of three, users should attempt to run the failed command in a non-Access Call batch environment in order to ascertain the cause of the failure.

It was stated earlier that a request could be a single DataManager command or a stream of DataManager commands, held in the input buffer (DINPUT). As the output from a request is placed in the user program buffer, DOUTPUT, one record at a time, it is necessary to have a mechanism by which the second and subsequent output lines or messages can be transferred to this buffer for processing. This mechanism is provided by the control field DFUNC. When servicing any request, the value of DFUNC is first set to indicate that the Access Call is supplying a request from DINPUT. The Call is then made; if successful, it leaves the first output record in DOUTPUT. After processing this record, the user program alters the value in DFUNC to indicate that the next output record is to be placed in DOUTPUT, and repeats the call. This process is repeated until all output records resulting from the request have been processed. Tests on another control field, DOUTWAIT, can determine how many output records are still waiting in the ControlManager buffer for transfer to DOUTPUT (but DOUTWAIT should not be tested alone for end of output, as this could leave a line or message unprocessed in DOUTPUT: DRETURN should be tested first, as stated in the previous paragraph).

The user can select, by setting a value in DOPTION5, whether the output records placed from the ControlManager buffer into DOUTPUT are to comprise all lines and messages output, or only the lines, or only the messages. The placing of lines or messages into DOUTPUT does not destroy their records in the ControlManager buffer.

The user can elect to process messages in these different ways:

- Process all messages first, and then proceed to process the output lines
- Process the output lines first, followed by the messages
- Process them together in the order in which they were output
- Switch between lines and messages at any time by resetting the values in DOPTION5 and DFUNC between calls.

The values set in other control fields can similarly be varied between calls to control the processing; the fields for which this applies have the remark `Must be set by user program in` their specification in "Storage Area Specifications" on page 33. The user should note, however, that many control fields are not examined on every Access Call; whether they are examined is dependent upon the value in DFUNC. Where this applies, this fact is also stated in the specification in "Storage Area Specifications" on page 33.

The storage allocated to the Transaction Work Area must be sufficient to contain these:

- DCONTROL (9112 bytes, plus the length of DINPUT as specified by DINPLEN, DINPLREC and DOPTION3)
- DOUTPUT (the size of which is specified by DOUTLEN)
- Any user-defined fields.

Installation And Job Control Requirements

For the linkage editing of user programs that utilize the Access Call capability under CICS, and for the job control requirements for running these programs, reference should be made to the DataManager-specific part of whichever of these manuals is appropriate to the user's environment:

- *ASG-Manager Products Installation in OS Environments*
- *ASG-Manager Products Installation in DOS Environments*.

Storage Area Specifications

The table below defines the individual fields of the three areas that a user must set up in their own Transaction Work Area in order to use the Access Call capability under CICS. ASG-Supplied Member Name means the name of the corresponding member of the User Interface Dictionary. See Chapter 4, "Output Formats," on page 41. Decimal Offset means the address offset from the start of the Transaction Work Area.

Fields Required in the User Program Work Area

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
Control parameter group, containing:	DCONTROL	0	GROUP	—	—	This field must immediately follow the Task Control Area. Length in bytes is 9112 plus the length of DINPUT.
ControlManager work area	DDMAREA	0	Character	9000		Reserved for ControlManager use.
Communication field	DCOMMUNE	9000	Character	64	—	Reserved for ControlManager use. Must be initialized by user program to binary zero/'LOW VALUES' before the first Access Call for any dictionary.
Reserved	DDMRLM	9064	—	4	—	Reserved.
Reserved	DDMRTS	9068	—	4	—	Reserved.
ControlManager output buffer size	DBUFFLEN	9072	—	4	Non-zero	Must be initialized by user program with the size in bytes of the temporary storage records to be created by ControlManager to hold output from the current Access Call request.
Input buffer size/Input line count	DINPLEN	9076	Binary	2	Non-zero (max. 32767)	Examined only when DFUNC=1. If DOPTION3=0, it specifies the byte size of the input buffer used. Its value should be a multiple of the value of DINPLREC. If DOPTION 3=1, it specifies the number of records in the input buffer used (where each record has length DINPLREC).

Fields Required in the User Program Work Area

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
Input line size	DINPLREC	9078	Binary	2	Non-zero (max. 255)	Must be set by user program with the size in bytes of the input lines. When DOPTION3=0, the size must not be greater than the value in DINPLEN and should divide exactly into the value in DINPLEN. DINPLREC is examined only when DFUNC=1.
Output record maximum size	DOUTLEN	9080	Binary	2	Non-zero (max. 600)	Must be set by user program with the maximum size of output record to be held in DOUTPUT. If the value specified is less than the maximum possible output record size, then output records that exceed the specified maximum size will be truncated from the right. If truncation of the record in DOUTPUT occurs, the original length of the output line or message is given in DOUTLREC. The maximum value required for DOUTLEN is 600. DOUTLEN is examined when DFUNC=1, 2, or 3.
Output record actual size	DOUTLREC	9082	Binary	2	—	Returned by ControlManager. Gives the original length in bytes of the output record placed in DOUTPUT. This length will be greater than the value in DOUTLEN if truncation of the output record has occurred.
Function required	DFUNC	9084	Numeric-character	1		Must be set by user program to indicate the function required by the current Access Call, thus:
					'1'	• This call is supplying a new request in DINPUT.
					'2'	• The next output line or message is to be placed in DOUTPUT.

Fields Required in the User Program Work Area

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
					'3'	• ControlManager is to reset to the first output line or message and place it in DOUTPUT.
Return code, contents of DOUTPUT	DRETURN	9085	Numeric-character	1		Returned by ControlManager. Indicates what has been placed in DOUTPUT, thus:
					'0'	• Content of DOUTPUT is a line
					'1'	• Content of DOUTPUT is a diagnostic message
					'2'	• Nothing has been placed in DOUTPUT
					'3'	• Nothing has been placed in DOUTPUT, and ControlManager has terminated
Not used	DRETURN1	9086	—	1	—	—
Return code, input condition	DRETURN2	9087	Numeric-character	1		Returned by ControlManager. Indicates the result of input processing, thus:
					'0'	• Input was processed
					'3'	• A serious error condition occurred, which caused ControlManager to terminate processing of the request
Output message severity code	DSEVRITY	9088	Character	1		Returned by ControlManager. Indicates the highest severity code of any message in the ControlManager output buffer, thus:
					Blank	• No Message Output
					'I'	• One or more informatory messages output

Fields Required in the User Program Work Area

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
					'W'	• One or more warning messages output
					'E'	• One or more error messages output
					'S'	• One or more serious messages output
					'C'	• A critical message output
Not used	DOPTION1	9089	—	1	—	—
Input option	DOPTION2	9090	Numeric-character	1		Must be set by user program to indicate input option, thus:
					'0'	• Read input only
					'3'	• Read and print input
						Examined only if DFUNC=1.
Input buffer size option	DOPTION3	9091	Numeric-character	1		Must be set by the user program to specify the use of DINPLEN:
					'0'	• DINPLEN holds the size in bytes of the input buffer/temporary storage record used
					'1'	• DINPLEN holds the number of records in the input buffer/temporary storage record used.
Not used	DOPTION4	9092	—	1	—	—
Output option	DOPTION5	9093	Numeric-character	1		Must be set by user program to indicate output option, thus:
					'0'	• Only output lines to be placed in DOUTPUT.

Fields Required in the User Program Work Area

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
					'1'	• Only diagnostic messages to be placed in DOUTPUT.
					'2'	• Output lines and messages to be placed in DOUTPUT in the order in which they were generated.
						Examined only if DFUNC=1 or 3.
Output format	DOPTION6	9094	Numeric-character	1		Must be set by user program to indicate the output format required, thus:
					'0'	• Lines in DOUTPUT to be in same format as POST/MAIL output.
					'1'	• Lines in DOUTPUT to be in normal DataManager output print format.
						Examined only if DFUNC=1.
Not used	DOPTION7	9095	—	1	—	—
Not used	DOPTION8	9096	—	1	—	—
Not used	DOPTION9	9097	—	1	—	—
Count of output records queued	DOUTWAIT	9098	Packed-decimal	3	five digits	Returned by ControlManager to indicate the number of output records still waiting in the ControlManager storage records to be placed in DOUTPUT.
Number of output lines generated	DOUTTOTL	9101	Packed-decimal	3	five digits	Returned by ControlManager when DFUNC=1, to indicate the number of output lines generated by the previous Access Call request.

Fields Required in the User Program Work Area

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
Number of message lines generated	DOUTTOTM	9104	Packed-decimal	3	five digits	Returned by ControlManager when DFUNC=1, to indicate the number of message lines generated by the previous Access Call request.
Command number	DINPNO	9107	Packed-decimal	2	1 - 999	Returned by ControlManager to indicate the command number of the command to which the output record relates. The commands are numbered from 1 in increments of 1. The command number is reset to 1 when an Access Call is made with DFUNC=1.
Reserved	DRESERVE	9109	—	3	—	Reserved for future use.
Input buffer	DINPUT	9112		See Note 1.		When DFUNC=1, must be set by user program to either: • The lines of DataManager input (each of the length specified by DINPLREC) to be processed by the current Access Call, or • The 8 character identification of the temporary storage record that holds the DataManager input.

Fields Required in the User Program Work Area

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
Output buffer: group, containing:	DOUTPUT	See Note 2	GROUP			This field is used by ControlManager to return output records resulting from the user's Calls. One record is returned by each Call. A record can be an output line or a diagnostic message. If the output line or message is shorter than the length of DOUTPUT (as specified in DOUTLEN), DOUTPUT is space-filled to the right. Chapter 4, "Output Formats," on page 41 describes how to ascertain the formats of output records.
Character of output	DOUTCHAR	—	Character	1	—	—

Notes

1 ControlManager assumes that the size of DINPUT is:

— 8 bytes, if DINPUT contains a temporary storage record identification

— (DINPLEN) bytes, if DOPTION3=0

— (DINPLEN*DINPLREC) bytes, if DOPTION3=1.

2 ControlManager assumes that DOUTPUT immediately follows DINPUT in the user's Transaction Work Area. Thus, the offset of DOUTPUT is the offset of DINPUT, plus the length of DINPUT as calculated according to Note 1.

4 Output Formats

This chapter outlines the description of the formats in which DataManager output is produced when the User Interface facility is used. The general format of each line of DataManager output is described in the *ASG-DataManager User's Guide*, and the layout of the output of individual DataManager commands is designed to be self-explanatory. However, users of the User Interface facility need explicit definitions of the relevant output formats to enable user programs to be correctly coded to process this output.

The formats to be described can be considered in two groups:

- The formats of the output text generated by individual DataManager commands
- The format of the records in which these texts are output for processing by user programs.

These two groups of formats are described as follows:

- The formats in which output records are generated are described in "Output Records" on page 42.
- The textual formats of the output of individual DataManager commands are described in a data dictionary whose members contain the definitions of each format. Users are also provided with datasets that contain COBOL, PL/I, Assembler, and MARK IV copies of these definitions. (These copies are not themselves data dictionary members; they are blocks of source code written in COBOL, PL/I, Assembler, or MARK IV (as appropriate) that can be included in user programs written in the corresponding programming languages.) A description of the formats is provided in this form for these reasons:
  - Providing a written description of the output format of each command would make this publication excessively bulky
  - Users can incorporate the COBOL, PL/I, Assembler, or MARK IV copies of these descriptions directly in user programs, for use as definitions of buffers that can accept the output of DataManager commands.

The data dictionary that provides these definitions is called the User Interface Dictionary; it is described in "The User Interface Dictionary" on page 44. The COBOL, PL/I, Assembler, and MARK IV copies of the Dictionary are respectively called the COBOL, PL/I, Assembler, and MARK IV Source Libraries; they are described in "The User Interface Source Libraries" on page 49. "The User Interface Datasets" on page 50 describes the datasets in which the User Interface Dictionary and the Source Libraries are supplied to users.

Output Records

The output records generated during the use of the User Interface facility are variable-length records. Each line of DataManager output is generated as one of these records.

Each record is generated in either a fixed or a variable format. Users can specify which format is to be used for the output records generated by individual DataManager commands, by using the NORMAL keyword of the POST/MAIL command (when using the POST/MAIL capability) or by varying the value of the DOPTION6 field of the Access Call Control Block (when using the Access Call capability).

The variable format is the format in which DataManager output records are normally printed or displayed (that is, when the User Interface facility is not used).

The fixed format is the format that is provided specially for the User Interface facility. It is designed to allow the convenient processing of output records by user-written programs. The fixed formats are the formats that are described in the User Interface Dictionary. Both fixed and variable format output records have this structure:

- A four-byte header. The first two bytes contain the total length of the output record (including the header). The second two bytes are not used. This header is included in output records that are generated by the POST/MAIL command, but not in records generated by the Access Call capability.
- A one-byte carriage control character. The binary value in this byte is the number of blank lines that would be printed before the text of the output record. A value of 255 (hexadecimal FF) denotes skip to head of form.
- A sequence of characters that comprises the text of the output record. The number of characters in the sequence is given by the contents of the record header minus five (in the case of POST/MAIL output records) or the value of the DOUTLREC control field minus one (in the case of Access Call output records).

In variable format records, the arrangement of characters reflects the text actually displayed or printed; each group of non-space characters is separated by one or more spaces, and thus the length of the sequence depends on the number of text characters in the line.

In fixed format records, each group of non-space characters in an output line is allocated a fixed number of bytes (for example, a field of 32 bytes is allocated wherever a dictionary member name can appear in an output line). Whenever the number of characters generated is less than the size of the field allocated to hold it, then the field is filled up with trailing spaces (or leading zeros, if the field only contains numeric characters). Each of these fields is separated in the sequence by a fixed number of space characters. This means that the length of the sequence is independent of the number of characters that the line contains each time that it is generated. Thus, in fixed format records, the length of each type of output line, and the length of each group of characters within a line, is predefined. These line lengths and character group lengths are stated for users in the User Interface Dictionary (see "The User Interface Dictionary" on page 44).

These figures show an example of a line of DataManager output held as a record in the two formats. The output line shown is the first line of the report of an ITEM dictionary member called ABC. (The text of this line is `REPORT OF ITEM ABC`.) Figure 6 shows the record structure when this line of output is generated as a variable format record. Figure 7 on page 44 shows the record structure when this line of output is generated as a fixed format record using the POST/MAIL capability. (If the record is generated using the Access capability, then it does not include the four-byte header.)

Figure 6 • Variable-format Output Record Containing a Report Heading

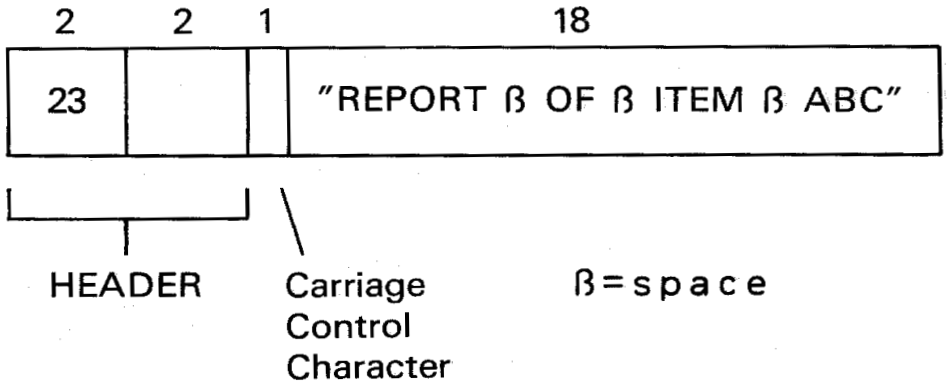
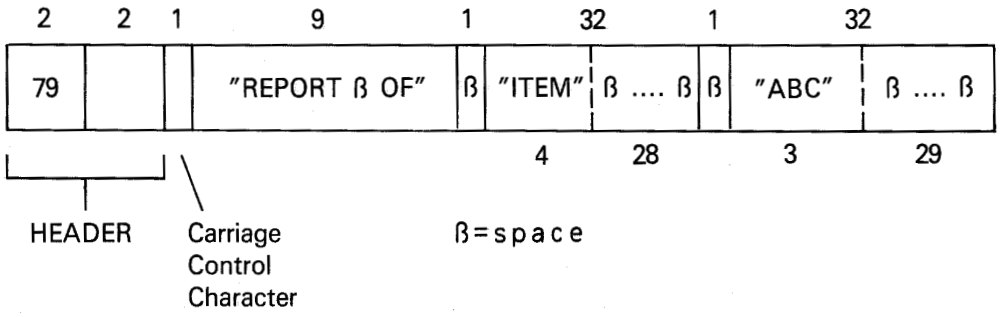


Figure 7 • Fixed-format Output Record Containing a Report Heading



The User Interface Dictionary

The User Interface Dictionary is a data dictionary that is supplied to users who purchase the User Interface Facility selectable unit (DMR-UI1). The members of this Dictionary are provided as part of the documentation of these Manager facilities:

- *ASG-DataManager User Interface*
- *ASG-DataManager User Defined Output*

The following description applies only to those members of the User Interface Dictionary that relate to the description of the User Interface facility. Members of the Dictionary that relate to other selectable units are described in the appropriate facility manuals.

The members of the User Interface Dictionary that relate to the User Interface facility describe the following:

- The fixed formats in which the User Interface Facility generates DataManager command output
- The structure of the Access Call Control Block (DCONTROL) which is used with the Access Call capability.

The various possible fixed output formats for each command are described by hierarchies of GROUP and ITEM members of the User Interface Dictionary. In principle there is a separate hierarchy of members for each command, although some ITEM members may be shared by several hierarchies in cases where parts of the output of different commands have a common format.

To each DataManager command there corresponds a GROUP member of the User Interface Dictionary; this GROUP member serves as the top-level member of the hierarchy for that command. This top member references, directly or indirectly, all the other members of the hierarchy for that command. Each member's definition either contains references to the members that describe the formats of smaller parts of the output or describes the format of some part of the command output.

As an example, consider the description of the format of the output produced by the SHOW ALIAS-TYPES command. There is a separate fixed format specified for each of the types of line output that a SHOW ALIAS-TYPES command may generate. These types are: a (header) line that displays the number of general aliases, header lines for the listing of specific alias-type keywords, and a line of output for each specific alias-type.

The User Interface Dictionary member named DSHA is the highest-level member of a hierarchy of members that provides a description of each of the possible fixed-length formats in which the output of a SHOW ALIAS-TYPES command can be generated. DSHA references other members in the hierarchy for SHOW ALIAS-TYPES by means of this clause:

```
CONTAINS  DSHASKP,
OSHAGEN  ELSE  OSHASPEC  ELSE  DSHAA  ELSE  OSHAB ELSE  OSHAEND
```

This clause specifies that the Dictionary member DSHASKP is common to the descriptions of all the SHOW ALIAS-TYPES output formats, since it describes the one-byte carriage control character that precedes the text of each DataManager output record (as described in "Output Records" on page 42). Each of the Dictionary members DSHAGEN, DSHASPEC, DSHAA, DSHAB, and DSHAEND describes the format of one of the types of output line mentioned above. DSHAA, for example, gives a description of the format of the line that holds a single alias-type (apart from the carriage control byte). This description is provided by the CONTAINS clause of the DSHAA member definition, which has this form:

```
CONTAINS (2)SPACE, DSHAANUM, (4)SPACE, DSHAANAM.
```

This defines the format of this line (apart form the initial one-byte header of the output record) as two spaces, followed by the alias-type number, four spaces, and the alias-type keyword. The format of each of these parts of the line is defined by means of further members, as follows. A member named SPACE is defined to hold a single space, a member named DSHAANUM defines the format of the alias-type number, and a member named DSHAANAM defines the format of the alias-type keyword. These three members are ITEM members of the dictionary and do not contain references to any other dictionary members. They are at the bottom of the hierarchy of members that defines the format of SHOW ALIAS-TYPES output.

The format of the output generated by other commands is defined in a similar way by other hierarchies of members within the User Interface Dictionary.

The members of the User Interface Dictionary are named in a systematic way so that each member's name indicates the command whose output it is helping to describe. The first three, four, or five letters of each member name denote the relevant command. For example, all members that contribute to the description of the output format of the SHOW ALIAS-TYPES command have DSHA as their first four letters. These groups of letters, and the commands to which each of them relates, are listed in the table below.

The structure of the Access Call Control Block is described in the User Interface Dictionary as the definition of a DataManager GROUP member (called DCONTROL). The form of each of the individual fields within the Control Block is also shown, as a set of ITEM members contained in DCONTROL. The names used for these members are the same as the names used in the Access Call Control Block table in Chapter 3, "The Access Call Capability," on page 11.

The User Interface Dictionary is supplied to users in a number of separate statuses. Each single status contains the version of the Dictionary that applies to a particular DataManager release. These statuses are currently supplied to users:

- Status POST1900 for DataManager Version 1 Release 9
- Status POST2400 for DataManager Version 1 Release 10
- Status POST2500 for DataManager Version 2 Release 5

Letter Prefixes for User Interface Dictionary Members

Prefixes	Output Formats Described
DBU	ANALYZE
DLA	LOG ANALYSIS
DLO	PRODUCE RECORD-LAYOUTS
DLS	LOG STATUS
DPA	REPORT ADABAS (see also DRPS and DRUA)
DPAAC	REPORT (in addition to DPR) when the report is on a SYSTEM, PROGRAM, or MODULE defined using the ADABAS Interface facility. DPAAC describes the PROCESSES part of the report
DPD	REPORT DL/I (IMS) (see also DRPS and DRUA)
DPF	Diagnostic Messages, PRINT, INSERT, ALTER, ADD, REPLACE, MODIFY, ENCODE
DPG	GLOSSARY (which also uses DPR)
DPJ	REPORT (in addition to DPR) when the report is on a SYSTEM, PROGRAM, or MODULE defined using the IMS (DL/I) Interface facility. DPJ describes the PROCESSES part of the report
DPL	LIST

Letter Prefixes for User Interface Dictionary Members

Prefixes	Output Formats Described
DPMTS	CONTROL LIST UDS
DPO	OWNER LIST
DPP	SECURITY LIST
DPQA	WHOSE ALIAS and WHAT IS
DPQB	WHAT BELONGS TO, WHAT FORMS and WHICH
DPQO	WHO
DPQU	WHAT USES and WHAT CONSTITUTES (DPQU is a group within in DPR)
DPR	REPORT (see also DRPS, DRUA, DRCM, DRFA, DRPA, DPAAC, and DPJ) and GLOSSARY for ALIAS, CATALOGUE, DESCRIPTION, and NOTE common clauses
DPS	LIST HISTORY
DPT	REPORT TOTAL (see also DRPS and DRUA)
DPU	SHOW UDS member types or attributes
DPUC	COMPARE UDS
DPY	REPORT SYSTEM2000-DATABASE and REPORT SCHEMA-RECORD (see also DRPS and DRUA)
DPYP	REPORT of a SYSTEM-2000 PROCESSES clause
DPYS	REPORT of a SYSTEM-2000-SUBSCHEMA-RECORD
DRA	REPORT of an ATTRIBUTE-TYPE member
DRCM	REPORT COMMAND-STREAM (see also DRPS and DRUA)
DRER	REPORT of an EXECUTIVE-ROUTINE member
DRET	REPORT of an ENTITY member
DRFA	REPORT and GLOSSARY of the GENERATION-CYCLE, RETENTION-PERIOD, GROWTH-RATE, DENSITY, VOLUME, and SIZE clauses of FILE members
DRFM	REPORT of a (DataManager or DesignManager) FORMAT member
DRG	REPORT of an ATTRIBUTE-GROUP member
DRGP	REPORT of a GLOBAL-PROFILE member
DRHR	REPORT of a HIERARCHY member
DRIB	REPORT of an INFOBANK-PANEL member

Letter Prefixes for User Interface Dictionary Members

Prefixes	Output Formats Described
DRID	REPORT IDMS DATA-BASE (see also DRPS and DRUA)
DRJA	REPORT IDMS AREA (see also DRES and DRUA)
DRJP	REPORT IDMS SUBSCHEMA (see also DRPS and DRUA)
DRJR	REPORT IDMS RECORD (see also DRPS and DRUA)
DRJS	REPORT IDMS SET (see also DRPS and DRUA)
DRJV	REPORT IDMS VIEW (see also DRPS and DRUA)
DRMG	REPORT of a MEMBER-TYPE-GROUP member
DRMT	REPORT of a MEMBER-TYPE member
DROD	REPORT of an OBSOLETE-DEFINITION member
DRPA	REPORT and GLOSSARY of AUTHOR, INSTALLATION, DATE-WRITTEN, SOURCE-COMPUTER, OBJECT-COMPUTER, SPECIAL-NAMES, 1-0- CONTROL, ASSIGNMENT, EDIT-INPUT, EDIT-OUTPUT and EDIT-UPDATE clauses of SYSTEM, PROGRAM and MODULE members
DRPF	REPORT of a LOGON-PROFILE member
DRPI	REPORT (in addition to DPR) when the report is on a SYSTEM, PROGRAM or MODULE defined using the ADABAS Interface facility. DRPI describes the PROCESSES part of the report.
DRPS	REPORT and GLOSSARY common clauses, except ALIAS, CATALOGUE, DESCRIPTION, and NOTE, which are in DPR
DRSA	REPORT of a SUDS-AREA member
DRSB	REPORT of a SUDS-SUBSCHEMA member
DRSD	REPORT of a SUDS-DATABASE member
DRSI	REPORT of a DL 1-AREA member
DRSP	REPORT of a SUDS-PROCESS member
DRSR	REPORT of a SUDS-RECORD member
DRSS	REPORT of a SUDS-SET member
DRSV	REPORT of a SUDS-VIEW member
DRUA	REPORT and GLOSSARY user defined attributes
DRUV	REPORT of a USERVIEW member

Letter Prefixes for User Interface Dictionary Members

Prefixes	Output Formats Described
DRVS	REPORT of a VIEWSET member
DSHA	SHOW ALIAS-TYPES
DSL	STATUS LIST

The User Interface Source Libraries

The command output format information that is provided by the User Interface Dictionary is also supplied in a number of further forms. Each of these is a copy of the User Interface written in a particular programming language. The languages in which these copies are supplied are COBOL, PL/I, Basic Assembler (BAL), and the MARK IV File Definition Language. These copies of the User Interface Dictionary are called User Interface Source Libraries. The COBOL User Interface Source Library, for example, contains the source code of a COBOL Data Division that corresponds to the contents of the User Interface Dictionary. Each of the Source Libraries is generated from the User Interface Dictionary by means of the DataManager Source Language Generation facility.

Users can incorporate the individual Source Libraries (in whole or in part) into their user programs. For example, a user who wishes to invoke the Access Call capability from a COBOL user program can include all or part of the COBOL User Interface Source Library in the program to serve as a buffer for holding the output of DataManager commands.

The data items included in each Source Library are named systematically, as follows. The first three, four, or five letters of each data item's name are the same as the initial letters of the name of the corresponding member of the User Interface Dictionary (as listed in the table above). The next three letters of each data item name denote the particular Source Library in which the data item is included. These letters are:

- COB for the COBOL User Interface Source Library
- PLI for the PL/I User Interface Source Library
- BAL for the Assembler User Interface Source Library
- MIV for the MARK IV User Interface Source Library.

Thus, for example, a user who wished to process the output from SHOW ALIAS-TYPES commands within a PL/I user program, would copy the definition of variables whose names begin with the letters DSHAPLI (and subordinate variables) into their program from the PL/I User Interface Source Library.

All the Source Libraries (except for the MARK IV User Interface Source Library) also include a definition of the Access Call Control Block. These definitions can also be directly incorporated in user programs. These are the names used in each case:

- DCONTCOB in the COBOL User Interface Source Library
- DCONTPLI in the PL/I User Interface Source Library
- DCONTBAL in the Assembler User Interface Source Library

The User Interface Datasets

The User Interface Dictionary and the various User Interface Source Libraries are supplied to users in the form of datasets on the Manager Products installation tape. These datasets are called User Interface Datasets. Their individual names and contents are listed in the table below.

User Interface Datasets

MP.UICOB	COBOL User Interface Source Library
MP.UIPLI	PL/I User Interface Source Library
MP. UIBAL	Assembler User Interface Source Library
MP. UIMIV	MARK IV User Interface Source Library

The installation of these datasets is described in *ASG-Manager Products Installation in OS Environments* and *ASG-Manager Products Installation in DOS Environments*.

5 The Input User Exit Capability

The Input User Exit capability enables users to examine and amend DataManager input records after they have been read but before they are executed by DataManager.

When the User Interface facility is supplied, the installation includes a module called the Input User Exit routine (DMEX1) which is entered from ControlManager after each DataManager input line has been read, but before it is executed. The ASG-supplied version of the DMEX1 routine does not carry out any processing of the input records. Users tailor the Input User Exit capability to carry out the processing of each input record that they require by amending the supplied version of DMEX1 with their own code (as described in "Tailoring the Capability" on page 57).

The user program (in the form of the DMEX1 routine) communicates with ControlManager via a control block (Input User Exit Control Block) that is passed as a parameter at each entry of the DMEX1 routine. The Input User Exit Control Block contains a number of fields, whose settings control the use of the capability. Some fields can be set by the user's coding within DMEX1, and others are set by ControlManager or DataManager.

The user can specify the action that ControlManager carries out on each input record by setting the values of particular fields of the Control Block. These actions include having ControlManager accept the current input record for subsequent execution by DataManager, rejecting the record, and issuing a DM09999 diagnostic message. The user specifies these actions by including coding in the DMEX1 routine to set the appropriate fields of the Input User Exit Control Block.

Other fields of the Input User Exit Control Block are set by ControlManager or DataManager to provide items of information for users. These items include the main storage location and the length of the current line of input. This information enables the user to amend the current input line before it is executed. Other items include the name and status of the currently accessed data dictionary (if any). Users can test the values of these fields by means of coding within DMEXl and use this information to control the processing performed within the DMEX1 routine.

Other fields of the Control Block are reserved for use by ControlManager alone, so that it can carry forward data from one entry of DMEX1 to the next. In addition, one field of the Control Block is provided for use by DMEXI coding to enable users to preserve selected data between successive entries of DMEXI.

The individual fields of the Input User Exit Control Block are shown in the table below. The use of each of the fields is described in the table, and the following sections describe some of the ways in which the capability can be used. In the Remarks column of each table entry, the phrases *Set by ControlManager* and *Set by DataManager* indicate that those fields are set by ControlManager or DataManager. The phrase *Set by user program* indicates those fields that users can set by means of DMEX1 coding.

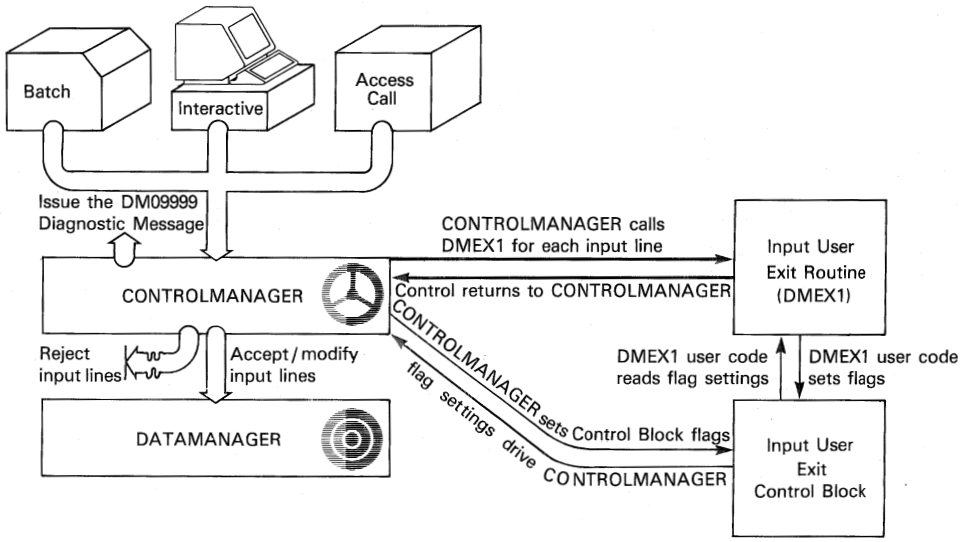
The DMEX1 routine is called once for each input record, and there is an additional call of DMEX1 after final input record processing is complete. (The final input record should normally be a LOGOFF command.) This final call gives users the opportunity to carry out any necessary termination processing, such as freeing any storage areas acquired by the user exit.

Figure 8 illustrates the role of the Input User Exit routine in the processing of each input record prior to its execution, and the way in which the user uses the capability by accessing and setting the fields of the Input User Exit Control Block.

A data dictionary does not need to be open when the Input User Exit capability is in use.

Using the Input User Exit capability does not impose any additional restrictions on the range of DataManager commands that are available to each user.

Figure 8 • The Input User Exit Capability



Input User Exit Control Block

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
—	DX1	0	GROUP	193	—	Group name for the whole Control Block
Address of current input record	DX1ADDR	0	Binary	4	—	Set by ControlManager
Length of current input record	DX1LEN	4	Binary	1		Set by ControlManager:
					Non-zero	• Record-length in bytes
					Zero	• After last Access Call of session, all input records have bee processed.
Dictionary name	DX1DICT	5	Character, left-justified, space-filled on the right	7		Set by DataManager. This field is spaces until the first DICTIONARY command has been accepted.
Dictionary user name	DX1USER	12	Character, left-justified, space-filled on the right	32	—	Set by DataManager. This field contains spaces until the first AUTHORITY command has been accepted. If the Controller's master password/Master Operator's password is accepted, this field contains MASTER/ OPERATOR.
Dictionary status name	DX1STAT	4	Character, left-justified, space-filled on the right	32	—	Set by DataManager. This field contains spaces until the first DICTIONARY command has been accepted; if there are no named statuses in the dictionary, it remains as spaces.
Status indicator	DX1FNF	76	Character	1		Set by DataManager. This field contains a space until the first DICTIONARY command has been accepted; if there are no named statuses in the dictionary, it remains as a space.
					'0'	• Status is non-frozen

Input User Exit Control Block

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
					'1'	• Status is frozen
					Space	• No named statuses in the dictionary
Return code	DX1RET	77	Character	1		Set by the user program:
					'0'	• Accept the input line.
					'1'	• Reject the input line.
					'2'	• Reject the input line but accept a substituted QUIT amendment line.
					'3'	• Reject the input line but accept a substituted line with a terminator in the first character position.
Reserved	DX1RES	78	Binary	2	—	Reserved for ControlManager use.
ControlManager save area address	DX1SAVE	80	Binary	4	—	Reserved for ControlManager use.
User exit work area	DX1WORK	84	Binary	8	—	Available to the user to pass indicators, addresses, and other parameters from one entry of DMEX1 to the next.
Message severity level	DX1SEV	92	Character	1	—	Set to a space by ControlManager. Set by the user program to any printable character if a message is to be output.
Message text	DX1MESS	93	Character	100		Set to spaces by ControlManager. When a user wishes to output the DM09999 diagnostic message, they should set this field to the message text to be output.
Dictionary user password	DX1PASS	193	Character, left-justified, space-filled on the right	8	—	Set by DataManager. This field contains spaces until the first AUTHORITY command is accepted.

Using the Capability

Setting Return Codes

After examining the current input record, the User Exit Module must set the return code field of the Control Block to indicate whether the record is to be accepted or rejected. If the return code indicates that the record is to be accepted, ControlManager proceeds to process its contents. Otherwise the next input record (if any) is read.

Generating a User Exit Diagnostic Message

Whether the current input record is accepted or rejected, the User Exit Module can pass to ControlManager the text and severity code of a diagnostic message to be output in respect to that record. The content of DX1SEV is always tested to determine whether such a message is to be output; if DX1SEV contains anything other than a space, then this message is given:

`DMO9999`*`x`* `USER EXIT MESSAGE:` *`variable`*

where:

`x` is the character held in DX1SEV

`variable` is the character string held in DX1MESS.

Amending Input Records

An input record can be amended before it is routed for execution, as follows. The return code for the record should be set to indicate that the record is to be accepted. If the length of an amended input record is greater than the length given in DX1LEN, the user must:

- Copy the record to another (user) area of main storage of sufficient size to hold the new length
- Process the record in that area of storage
- Provide the address of the amended record, and its length, in the control fields DX1ADDR and DX1LEN respectively.

Rejecting Input Records

The effects of rejecting an input record depend partly on the type of record rejected, as discussed in the following paragraphs. For this reason, three different values of the return code field (DX1RET) of the Input User Exit Control Block are provided for rejecting an input record; the circumstances in which they should be used are suggested in the following paragraphs. These return codes enable the Input User Exit routine to instruct ControlManager to:

- Reject the input record (return code=1)
- Reject the input record and substitute a replacement QUIT amendment record (return code=2)
- Reject the input record and substitute a replacement record containing a terminator in the first character position (return code=3)

If the Input User Exit routine rejects the second or any subsequent record of a command, and the contents of the records that are not rejected form a valid command, then that command is processed. If the records that are not rejected do not form a valid command, then appropriate diagnostic messages are output by ControlManager. If the rejected records include the command's terminator, the following command may also be treated as incorrect syntax. To avoid this, if ever a second or subsequent record of a command is rejected and the record includes the command's terminator, the Input User Exit routine should set DX1RET to 3, thus forcing the inclusion of a terminator.

Similarly, if any record of a command is rejected, and that record does not include a terminator, the Input User Exit routine can, if so desired, set DX1WORK to signal rejection of all subsequent input records, with a return code of 1, until a record with a terminator is received, when a code of 3 can be returned and the indicator in DX1WORK can be cleared. This avoids the need for further checking of the syntax of the remaining part of the command, avoids possible output of diagnostic error messages, and avoids the need for further validation of the input record by the Input User Exit routine.

If a rejected command, or a command whose terminator is forced by a return code of 3 is an ADD, INSERT, ALTER, or MODIFY command, then all the input records of the associated data definition or amendment can be rejected in a similar way.

If one of the records of a data definition associated with an ADD or INSERT command is rejected, only the accepted records of the data definition are written to the source dataset. The input record that contains the terminator of the data definition should never be rejected unless the whole ADD or INSERT command and the associated data definition are rejected.

If an input record rejected by the Input User Exit routine is an amendment line, then the ALTER or MODIFY command does not detect the rejection as an error condition; so that if none of the amendment lines accepted by the Input User Exit routine gives rise to an error condition and no QUIT amendment line is input, then the amended record is written to the source dataset. (See the specifications of the ALTER and MODIFY commands in the *ASG-Manager Products Dictionary/Repository User's Guide*.) If the Input User Exit routine rejects an amendment line, then the user should decide whether the Routine should subsequently:

- Reject all subsequent amendment line records until the amendment terminator or QUIT is read, or
- Process subsequent amendment line records, but force a QUIT when the amendment terminator is read (by a return code of 2), or
- Carry out the amendments resulting from those amendment lines that were not rejected (by passing a return code of 0 when the amendment terminator is read).

Tailoring the Capability

In order to make use of the Input User Exit capability, a user must include his or her own coding within the DMEX1 Input User Exit routine. This is done by editing, assembling, and link editing a new version of DMEX1, in order to replace the ASG-supplied version of DMEX1 with a version that contains the user's own coding.

To enable DMEX1 to be amended by users, its source code is supplied on the Installation Tape as part of the MP.SOURCE dataset (in the case of OS environments) or as part of the Source Statement Library (in the case of DOS environments). A new version of DMEX1 must be written in Basic Assembler, be in re-entrant code, follow the standard linkage conventions, and adhere to the register conventions used in the ASG-supplied version.

The job control statements needed for the assembly and link editing of an amended version of DMEX1 are given in the DataManager-specific part of the appropriate Manager Products installation manual.

6 The Access Call Capability Under CICS

For a general introduction to the Access Call capability, refer to the introduction in Chapter 3, "The Access Call Capability," on page 11.

Figure 9 on page 60 and Figure 10 on page 60 illustrate how the Access Call capability operates in CICS environments. Figure 9 on page 60 shows the way that control passes between a user program, ControlManager and DataManager. Figure 10 on page 60 shows the way in which DataManager input records and the corresponding generated output records are routed between a user program, ControlManager and DataManager.

Programming Access Calls Under CICS

User Programming

Under CICS, an Access Call is carried out by means of a CICS link to program DMGRDMRU. The Access Call can be coded as a macro level call of this form:

```
DFHPC TYPE=LINK.PROGRAM=DMGRDNRU
```

or as a command level call of this form:

```
EXEC CICS LINK PROGRAM('DMGRDMRU')
```

DMGRDMRU corresponds to the interface routine DMRUS which is used in non-CICS environments. When DataManager has completed processing the Access Call, control is returned to the user program.

At any stage of its execution, the user program may re-enter CICS between Access Calls (for example, to read input from the terminal). All the ControlManager and DataManager-related storage areas that need to be preserved for subsequent Access Calls are written to temporary storage records and retrieved as necessary.

ControlManager uses three storage areas, which must be defined to follow the Task Control Area (TCA) as the first fields in the user's Transaction Work Area (TWA). Any other user defined fields in the Transaction Work Area must follow these first three areas.

Figure 9 • The Access Call Capability under CICS (Control Flow)

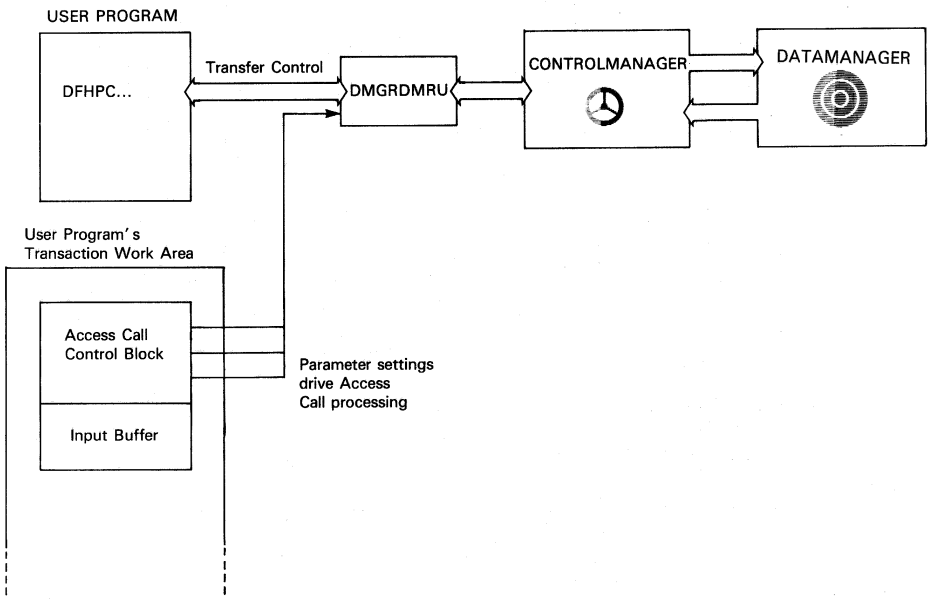
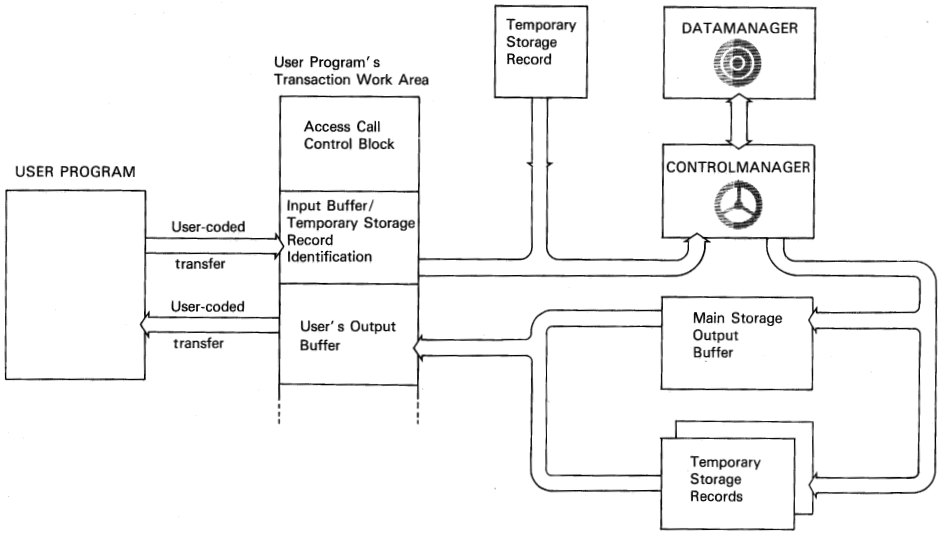


Figure 10 • The Access Call Capability under CICS (Data Flow)



The first of these storage areas is a control block (Access Call Control Block). This block contains a number of control fields (as defined in the table in "Storage Area Specifications" on page 64). The second area is an input buffer. The input buffer can be filled by the user program at run time with DataManager input lines. When the Access Call is performed, all the input in the buffer is processed as a single request to DataManager. A request can be a simple request to process one command only, or a complex request to process a stream of commands.

As an alternative to having input commands held in the above-mentioned buffer in the Transaction Work Area, users can specify that DataManager input is to be read from a temporary storage record. In this case, the input buffer in the Transaction Work Area contains only the eight-character identification of the temporary storage record that contains the input. The User Interface Facility decides which of these two areas contains the current input by comparing the first four characters of the Transaction Work Area input buffer with the four-character identification of the terminal that is associated with the user's transaction; if these are the same, then input is assumed to be held in a temporary storage record whose identification is given by the first eight characters in the Transaction Work Area input buffer, otherwise the input buffer itself is assumed to hold the input. If a temporary storage record is used to hold input, then the storage record is released when the input has been read.

The third of these storage areas receives each output record returned to the user program.

Any symbolic names allowed in the programming language in which the user program is coded can be used to name these three user program storage areas. In this publication, these names are used:

- DCONTROL for the Access Call Control Block
- DINPUT for the input buffer
- DOUTPUT for the output buffer

In addition to the description of the structure of the Access Call Control Block in this chapter, the structure of the Access Call Control Block is also described in a machine-readable form. This second description takes the form of a series of interrelated data dictionary member definitions. These dictionary members form part of the User Interface Dictionary which is supplied to User Interface facility users on their installation tape. Several programming language copies of these definitions are also included on installation tapes, in order to allow the inclusion of the appropriate copy into user programs. This second description of the Access Call Control Block, and a general description of the information provided by the contents of the User Interface Dictionary, is given in Chapter 4, "Output Formats," on page 41.

A definition of the last field in the DCONTROL data structure, DINPUT is not given, because its length depends on user requirements. Users must therefore define DINPUT themselves.

Specifications of the individual fields of the three named areas, DCONTROL, DINPUT, and DOUTPUT are given in "Storage Area Specifications" on page 64. The field names there used are those used for the corresponding data definitions in the User Interface Dictionary. Other names can be substituted in user programs if required.

The first command processed by the first Access Call of a session should be a LOGON or a DICTIONARY command. If it is a LOGON command, then the user is logged on to ControlManager under the logon identifier specified. If it is not a LOGON command, then the user is logged on to ControlManager by the Autolog process and the specified data dictionary is opened.

The last command processed by the final Access Call of a session should be a LOGOFF command. This command closes any open data dictionary and datasets, and logs the user off from ControlManager. One of the fields of the Access Call Control Block (DRETURN) has a value of three whenever a LOGOFF command has been processed.

For upwards compatibility, users may continue to use an ENDDMR command instead of LOGOFF as the final command. DRETURN is also set to three after an ENDDMR command is processed.

All DataManager command syntax rules apply equally to commands issued via Access Call. Terminators must be included, both for commands and for the data definitions or amendments associated with any INSERT, ADD, ALTER, MODIFY, or REPLACE commands. The whole of any such command and its associated data definition or amendment must be contained in a single Access Call.

Control Block Settings

The details of initializing or setting fields to control the various options available within Access Call are given in the specifications of the storage areas in "Storage Area Specifications" on page 64. If a field is defined as initialized by the user program, its value must be set by the user program for the first call to operate on a particular dictionary, and must not be altered by the user program until that dictionary is closed. If a field is defined as set by the user program, it can be reset for later calls. User programs must not modify the contents of any field of DCONTROL unless the specification of the field states that the contents are to be set or initialized.

When a request is processed by an Access Call, the DataManager output is placed first in a ControlManager buffer. The size of this buffer is determined by the control field DBUFFLEN. From this ControlManager buffer, the Access Call returns one output record at a time to the user program buffer DOUTPUT. An output record can be a line output by a DataManager command, or a message as defined in the *ASG-Manager Products Messages* manual or the Restricted Messages manual. A control field, DRETURN, informs the user program whether the content of DOUTPUT is a line or a message, or whether nothing is returned. In the latter case, tests on the DRETURN2 control fields can ascertain the reason. When control is returned to the user program after an Access Call, DRETURN should be the first field examined, followed if necessary by DRETURN2. If DRETURN2 shows a return code of three, users should attempt to run the failed command in a non-Access Call batch environment in order to ascertain the cause of the failure.

It was stated earlier that a request could be a single DataManager command or a stream of DataManager commands, held in the input buffer (DINPUT). As the output from a request is placed in the user program buffer, DOUTPUT, one record at a time, it is necessary to have a mechanism by which the second and subsequent output lines or messages can be transferred to this buffer for processing. This mechanism is provided by the control field DFUNC. When servicing any request, the value of DFUNC is first set to indicate that the Access Call is supplying a request from DINPUT. The Call is then made; if successful, it leaves the first output record in DOUTPUT. After processing this record, the user program alters the value in DFUNC to indicate that the next output record is to be placed in DOUTPUT, and repeats the call. This process is repeated until all output records resulting from the request have been processed. Tests on another control field, DOUTWAIT, can determine how many output records are still waiting in the ControlManager buffer for transfer to DOUTPUT (but DOUTWAIT should not be tested alone for end of output, as this could leave a line or message unprocessed in DOUTPUT: DRETURN should be tested first, as stated in the previous paragraph).

The user can select, by setting a value in DOPTION5, whether the output records placed from the ControlManager buffer into DOUTPUT are to comprise all lines and messages output, or only the lines, or only the messages. The placing of lines or messages into DOUTPUT does not destroy their records in the ControlManager buffer.

The user can elect to process messages in these different ways:

- Process all messages first, and then proceed to process the output lines
- Process the output lines first, followed by the messages
- Process them together in the order in which they were output
- Switch between lines and messages at any time by resetting the values in DOPTION5 and DFUNC between calls

The values set in other control fields can similarly be varied between calls to control the processing; the fields for which this applies have the remark `Must be set by user program` in their specification, see section below. The user should note, however, that many control fields are not examined on every Access Call; whether they are examined is dependent upon the value in DFUNC. Where this applies, this fact is also stated in the section below.

The storage allocated to the Transaction Work Area must be sufficient to contain these:

- DCONTROL (9112 bytes, plus the length of DINPUT as specified by DINPLEN, DINPLREC and DOPTION3)
- DOUTPUT (the size of which is specified by DOUTLEN)
- Any user-defined fields.

Installation And Job Control Requirements

For the linkage editing of user programs that utilize the Access Call capability under CICS, and for the job control requirements for running these programs, refer to the DataManager-specific part of the appropriate publication:

- *ASG-Manager Products Installation in OS Environments*
- *ASG-Manager Products Installation in DOS Environments*.

Storage Area Specifications

The table below defines the individual fields of the three areas that a user must set up in their own Transaction Work Area in order to use the Access Call capability under CICS. ASG-Supplied Member Name means the name of the corresponding member of the User Interface Dictionary. See Chapter 4, "Output Formats," on page 41. Decimal Offset means the address offset from the start of the Transaction Work Area.

Fields Required in the User Program Work Area

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
Control parameter group, containing:	DCONTROL	0	GROUP	—	—	This field must immediately follow the Task Control Area. Length in bytes is 9112 plus the length of DINPUT.
ControlManager work area	DDMAREA	0	Character	9000		Reserved for ControlManager use.
Communication field	DCOMMUNE	9000	Character	64	—	Reserved for ControlManager use. Must be initialized by user program to binary zero/'LOW VALUES' before the first Access Call for any dictionary.
Reserved	DDMRLM	9064	—	4	—	Reserved.
Reserved	DDMRTS	9068	—	4	—	Reserved.
ControlManager output buffer size	DBUFFLEN	9072	—	4	Non-zero	Must be initialized by user program with the size in bytes of the temporary storage records to be created by ControlManager to hold output from the current Access Call request.
Input buffer size/Input line count	DINPLEN	9076	Binary	2	Non-zero (max. 32767)	Examined only when DFUNC=1. If DOPTION3=0, it specifies the byte size of the input buffer used. Its value should be a multiple of the value of DINPLREC. If DOPTION 3=1, it specifies the number of records in the input buffer used (where each record has length DINPLREC).

Fields Required in the User Program Work Area

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
Input line size	DINPLREC	9078	Binary	2	Non-zero (max. 255)	Must be set by user program with the size in bytes of the input lines. When DOPTION3=0, the size must not be greater than the value in DINPLEN and should divide exactly into the value in DINPLEN. DINPLREC is examined only when DFUNC=1.
Output record maximum size	DOUTLEN	9080	Binary	2	Non-zero (max. 600)	Must be set by user program with the maximum size of output record to be held in DOUTPUT. If the value specified is less than the maximum possible output record size, then output records that exceed the specified maximum size will be truncated from the right. If truncation of the record in DOUTPUT occurs, the original length of the output line or message is given in DOUTLREC. The maximum value required for DOUTLEN is 600. DOUTLEN is examined when DFUNC=1, 2, or 3.
Output record actual size	DOUTLREC	9082	Binary	2	—	Returned by ControlManager. Gives the original length in bytes of the output record placed in DOUTPUT. This length will be greater than the value in DOUTLEN if truncation of the output record has occurred.
Function required	DFUNC	9084	Numeric-character	1		Must be set by user program to indicate the function required by the current Access Call, thus:
					'1'	• This call is supplying a new request in DINPUT.
					'2'	• The next output line or message is to be placed in DOUTPUT.

Fields Required in the User Program Work Area

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
					'3'	• ControlManager is to reset to the first output line or message and place it in DOUTPUT.
Return code, contents of DOUTPUT	DRETURN	9085	Numeric-character	1		Returned by ControlManager. Indicates what has been placed in DOUTPUT, thus:
					'0'	• Content of DOUTPUT is a line.
					'1'	• Content of DOUTPUT is a diagnostic message.
					'2'	• Nothing has been placed in DOUTPUT.
					'3'	• Nothing has been placed in DOUTPUT, and ControlManager has terminated.
Not used	DRETURN1	9086	—	1	—	—
Return code, input condition	DRETURN2	9087	Numeric-character	1		Returned by ControlManager. Indicates the result of input processing, thus:
					'0'	• Input was processed.
					'3'	• A serious error condition occurred, which caused ControlManager to terminate processing of the request.
Output message severity code	DSEVRITY	9088	Character	1		Returned by ControlManager. Indicates the highest severity code of any message in the ControlManager output buffer, thus:
					Blank	• No Message Output
					'I'	• One or more informatory messages output

Fields Required in the User Program Work Area

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
					'W'	• One or more warning messages output
					'E'	• One or more error messages output
					'S'	• One or more serious messages output
					'C'	• A critical message output
Not used	DOPTION1	9089	—	1	—	—
Input option	DOPTION2	9090	Numeric-character	1		Must be set by user program to indicate input option, thus:
					'0'	• Read input only
					'3'	• Read and print input
						Examined only if DFUNC=1.
Input buffer size option	DOPTION3	9091	Numeric-character	1		Must be set by the user program to specify the use of DINPLEN:
					'0'	• DINPLEN holds the size in bytes of the input buffer/temporary storage record used.
					'1'	• DINPLEN holds the number of records in the input buffer/temporary storage record used.
Not used	DOPTION4	9092	—	1	—	—
Output option	DOPTION5	9093	Numeric-character	1		Must be set by user program to indicate output option, thus:
					'0'	• Only output lines to be placed in DOUTPUT

Fields Required in the User Program Work Area

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
					'1'	• Only diagnostic messages to be placed in DOUTPUT
					'2'	• Output lines and messages to be placed in DOUTPUT in the order in which they were generated.
						Examined only if DFUNC=1 or 3.
Output format	DOPTION6	9094	Numeric-character	1		Must be set by user program to indicate the output format required, thus:
					'0'	• Lines in DOUTPUT to be in the same format as POST/MAIL output
					'1'	• Lines in DOUTPUT to be in the normal DataManager output print format.
						Examined only if DFUNC=1.
Not used	DOPTION7	9095	—	1	—	—
Not used	DOPTION8	9096	—	1	—	—
Not used	DOPTION9	9097	—	1	—	—
Count of output records queued	DOUTWAIT	9098	Packed-decimal	3	five digits	Returned by ControlManager to indicate the number of output records still waiting in the ControlManager storage records to be placed in DOUTPUT.
Number of output lines generated	DOUTTOTL	9101	Packed-decimal	3	five digits	Returned by ControlManager when DFUNC=1, to indicate the number of output lines generated by the previous Access Call request.

Fields Required in the User Program Work Area

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
Number of message lines generated	DOUTTOTM	9104	Packed-decimal	3	five digits	Returned by ControlManager when DFUNC=1, to indicate the number of message lines generated by the previous Access Call request.
Command number	DINPNO	9107	Packed-decimal	2	1 - 999	Returned by ControlManager to indicate the number of the command to which the output record relates. The commands are numbered from 1 in increments of 1. The command number is reset to 1 when an Access Call is made with DFUNC=1.
Reserved	DRESERVE	9109	—	3	—	Reserved for future use.
Input buffer	DINPUT	9112		See Note 1.		When DFUNC=1, must be set by user program to either: • The lines of DataManager input (each of the length specified by DINPLREC) to be processed by the current Access Call • The 8 character identification of the temporary storage record that holds the DataManager input

Fields Required in the User Program Work Area

Field Contents	ASG-Supplied Member Name	Decimal Offset	Form Description	Length in Bytes	Values	Remarks
Output buffer: group, containing:	DOUTPUT	See Note 2	GROUP			This field is used by ControlManager to return output records resulting from the user's Calls. One record is returned by each Call. A record can be an output line or a diagnostic message. If the output line or message is shorter than the length of DOUTPUT (as specified in DOUTLEN), DOUTPUT is space-filled to the right. Chapter 4, "Output Formats," on page 41 describes how to ascertain the formats of output records.
Character of output	DOUTCHAR	—	Character	1	—	—

Notes

1 ControlManager assumes that the size of DINPUT is:

— 8 bytes, if DINPUT contains a temporary storage record identification

— (DINPLEN) bytes, if DOPTION3=0

— (DINPLEN*DINPLREC) bytes, if DOPTION3=1.

2 ControlManager assumes that DOUTPUT immediately follows DINPUT in the user's Transaction Work Area. Thus, the offset of DOUTPUT is the offset of DINPUT, plus the length of DINPUT as calculated according to Note 1.

Index

A

Access Call 2, 11, 27, 59
- capability 2, 11, 59
- Control Block 13, 29–31, 42, 44, 61–62
  - Settings 15
- interface routines 11–12, 28, 59
- processing of input by 13
- programming of 11
- Work File 25
  - return code 20

ADD command 14, 31, 56, 62
ALTER command 14, 31, 56, 62
Amendments 57
AND keyword 7–8
Assembler 13, 41
- User Interface Source Library 41, 49

AUTHORITY command 53–54
Autolog process 14, 30, 62

B

BREAK key 21
Buffers
- User program alternative input (Access Call) 13

C

CALL statement 2, 13, 15, 27
capabilities
- outline of 1

CICS 2, 6, 10–11, 27, 59
- Destination Control Table 6
- Extrapartition Dataset 6

CLOSE command 5, 9
CMS 2
COBOL 13, 41
- User Interface Source Library 49

Command level call (CICS) 28, 59
Command number (Access Call) 26
ControlManager
- commands usable via the Facility 1

CREATE command 5

D

Data descriptions
- of output records 15, 30, 42, 62
- of parameter areas 16, 31, 62

DataManager
- commands usable with Access Call capability 11
- commands usable with Input User Exit capability 52
- commands usable with POST/MAIL capability 5

DBUFFLEN 18, 31, 33, 63, 65
DCOMMUNE 17, 33, 65
DCONTBAL 50
DCONTCOB 50
DCONTPLI 50
DCONTROL 14, 16–17, 30–33, 61–62, 64–65
DDMAREA 33, 65
DDMR 18
DDMRLM 33, 65
DDMRTS 33, 65
DFUNC 15, 20, 31, 34, 63, 66
Diagnostic messages (Input User Exit) 2, 51, 56
Dictionaries
- use with Input User Exit capability 52
- use with POST/MAIL capability 5, 9

DICTIONARY command 5–6, 9, 14, 53
DINPLEN 19, 32–33, 39, 64–65, 71
DINPLREC 19, 32, 34, 39, 64, 66, 71
DINPNO 26, 38, 70
DINPUT 15, 26, 30–31, 38–39, 61, 63, 70–71
DM.UIBAL dataset 50
DM.UICOB dataset 50
DM.UIMIV dataset 50
DM.UIPLI dataset 50
DMEX1 routine 51–52, 54
DMGRDMRU interface routine 28, 59
DMRUS 13
DMRUS interface routine 28, 59

DOPTION1 16, 22, 36, 68
DOPTION2 16, 23, 36, 68
DOPTION3 24, 32, 36, 39, 64, 68, 71
DOPTION4 24, 36, 68
DOPTION5 24, 32, 36, 63, 68
DOPTION6 25, 37, 42, 69
DOPTION7 25, 37, 69
DOPTION8 25, 37, 69
DOPTION9 25, 37, 69
DOPTIONS 15
DOS 2, 6, 10, 57
DOUTCHAR 27, 39, 71
DOUTLEN 19, 32, 34, 64, 66
DOUTLREC 20, 34, 66
DOUTPUT 14–15, 27, 30–32, 39, 61, 63–64, 71
DOUTTOTL 26, 37, 69
DOUTTOTM 26, 38, 70
DOUTWAIT 15, 26, 31, 37, 63, 69
DRESERVE 26, 38, 70
DRETURN 14–15, 20, 30–31, 35, 62–63, 67
DRETURN1 20, 35, 67
DRETURN2 15, 21, 31, 35, 63, 67
DSEVRITY 22, 35, 67
DX1 53
DX1ADDR 53, 55
DX1DICT 53
DX1FNF 53
DX1LEN 53, 55
DX1MESS 54
DX1PASS 54
DX1RES 54
DX1RET 54, 56
DX1SAVE 54
DX1SEV 54–55
DX1STAT 53
DX1USER 53
DX1WORK 54, 56

E

ENDDMR command 14, 30, 62
environments
  for using User Interface Facility 2
Examples
  of output record formats 43
  of POST/MAIL commands 8
  of use of User Interface Dictionary 45
Extrapartition 6

F

Fixed format output records 9, 42
FORTRAN 13

H

Hard Copy Feature 10

I

IMS/DC 2
Initializing of Access Call Control Block
  parameters 15, 29, 60
Input (Access Call) 12, 28, 59
  record size 13, 33, 65
  source 23, 29, 61
Input User Exit capability 2, 51
Input User Exit Control Block 51
INSERT command 14, 31, 56, 62

J

Job Control
  for Access Call capability 16, 32, 64
  for Input User Exit capability 57
  for POST/MAIL capability 9

L

LOGOFF command 14, 30, 62
LOGON command 14, 30, 62

M

Macro level call 28, 59
MAIL keyword 5, 7
MARK IV 41
  File Definition Language 49
  User Interface Source Library 49–50
Message severity mode 35, 54, 67
MODIFY command 14, 31, 56, 62
MP.SOURCE dataset 57
MPIN dataset 13, 23
MPOUT dataset 7, 22
Multiple dictionaries 16

N

NORMAL keyword 6, 8, 42

O

ONTO keyword 6, 8–9
OS 2, 6, 57
Output
  from Access Calls 2, 11, 15, 32, 42, 63
  from POST/MAIL commands 1, 8, 42
  messages (Access Call) 22, 32, 63
  messages (Input User Exit) 51, 54
  messages severity code 22, 35, 67
  records (Access Call) 15, 32, 63

P

PA1 key 25

Parameters
(Input User Exit) 53
Access Call Control Block 13, 17, 33, 65
sharing of, between dictionaries 16
PERFORM command 5–6
PL/I 13, 41
Optimizer 13
User Interface Source Library 49–50
POST keyword 5, 7
POST/MAIL
capability 5
command 5
Sequential Output Dataset 5, 9
POST/MAIL Sequential Output Dataset
format of 9
naming of 9
Primary output device 6–8, 10
Print option (Access Call) 22
Program testing 16

Q

QUIT amendment line 57

R

RELOAD command 5
REPLACE command 14, 31, 62
ROSCOE 2

S

Security 11
Setting of fields 13
Severity code 22, 54
SYSIPT dataset 13, 23
SYSLST dataset 7, 22
System Administrator 11

T

Task Control Area (CICS) 28, 33, 60, 65
Temporary Output Records 28, 59
Terminators 6, 14, 31, 56, 62
Testing of parameter fields 15
Transaction Work Area (CICS) 28, 32–33, 39, 60, 64, 71
TSO 2, 25
TSO/ISPF 2

U

User Defined output facility 44
User Defined syntax facility 1
User Interface Datasets 50
User Interface Dictionary 14, 17, 42
naming of data descriptions 46
statuses of 46
User Interface Source Libraries 49
naming of data descriptions 49
User programs, definition of 3
Uses of the Facility 1

V

Variable format output records 42