Publication Number
SH20-6597-01

IMS Application Development Facility II Version 2 Release 2

Data Dictionary Extension User's Guide

IBM

File Number
S/370/4300-32

Program Number
5665-348

IMS Application Development Facility II Version 2 Release 2

Data Dictionary Extension User's Guide

Publication Number
SH20-6597-01

File Number
S/370/4300-32

Program Number
5665-348

Second Edition (June 1986)

This edition applies to Version 2, Release 2 of the program product IMS Application Development Facility II (5665-348), and to all subsequent releases and modifications unless otherwise indicated in new editions or Technical Newsletters.

PREFACE

This publication provides information required to use the IBM DB/DC Data Dictionary in the IMS Application Development Facility II Version 2 Release 2 environment. It is not a stand-alone document and must be used in conjunction with the IBM DB/DC Data Dictionary library.

This manual consists of five chapters and five appendixes.

- **Chapter 1, "Overview"** gives a general background on the Data Dictionary Extension, its purpose and benefits.
- **Chapter 2, "The Model"** defines the IMSADF II Rules Generator operands to be added to the DB/DC Data Dictionary.
- **Chapter 3, "ADFIN Processor"** describes an IMSADF II method for adding or updating information in the categories and relationships of the Model using Rules Generator source as the original input.
- **Chapter 4, "ADFOUT Processor"** provides information for accessing the data contained in the DB/DC Data Dictionary data bases that make up the IMSADF II Data Dictionary Extension model.
- **Chapter 5, "Messages"** lists all the messages that can be generated when using the Data Dictionary Extension. The explanation, system action, and operator response are given for each message.
- The appendixes include:

  Appendix A, "ADFIN Processor Sample Procedure Output"

  Appendix B, "ADFOUT Processor Sample Procedure Output"

  Appendix C, "ADFIN Processor Module Definition"

  Appendix D, "ADFOUT Processor Module Definition"

  Appendix E, "IMSADF II STRTYPE"

RELATED PUBLICATIONS

IMSADF II PUBLICATIONS

- *IMS Application Development Facility II Version 2 Release 2 General Information*, GH20-6591.
- *IMS Application Development Facility II Version 2 Release 2 User Reference*, SH20-6592.
- *IMS Application Development Facility II Version 2 Release 2 Installation Guide*, SH20-6593.
- *IMS Application Development Facility II Version 2 Release 2 Application Development Reference*, SH20-6594.
- *IMS Application Development Facility II Version 2 Release 2 Application Development Guide*, SH20-6595.
- *IMS Application Development Facility II Version 2 Release 2 Rules Documentation User's Guide*, SH20-6596.
- *IMS Application Development Facility II Version 2 Release 2 Data Dictionary Extension User's Guide*, SH20-6597.
- *IMS Application Development Facility II Version 2 Release 2 Master Index*, SH20-6599.
- *IMS Application Development Facility II Version 2 Release 2 Introduction to Using the Interactive ADF*, SH20-6601.
- *IMS Application Development Facility II Version 2 Release 2 Interactive ADF Administration Guide*, SH20-6602.
- *IMS Application Development Facility II Version 2 Release 2 DATABASE 2 Application Specification Guide*, SH20-6603.
- *IMS Application Development Facility II Version 2 Release 2 Diagnosis Guide*, LY20-6401.

OTHER PUBLICATIONS

- *DB/DC Data Dictionary Administration and Customization Guide*, SH20-9174
- *DB/DC Data Dictionary Terminal User's Guide and Command Reference*, SH20-9189
- *DB/DC Data Dictionary Application Guide*, SH20-9190
- *DB/DC Data Dictionary Interactive Display Forms Facility User's Guide*, SR20-4726

CONTENTS

FIGURES

CHAPTER 1. OVERVIEW

The IMSADF II Data Dictionary Extension allows IMSADF II users to interface with the DB/DC Data Dictionary. The DB/DC Data Dictionary is an IBM Program Product used as a development tool to manage information about an installation's data processing resources. It is used to:

- Store and retrieve information about data and programs under development
- Retrieve information about existing data and programs

The Data Dictionary can be used to control and document the IMSADF II Master Rules. This concept is explained below. An installation that is using the IMSADF II master rule approach for setting up their application development environment will find that it can easily generate the following Rules Generator source statements directly from their Data Dictionary via the ADFOUT processor:

- IMSADF II master rules source (IMSADF II data base SEGMENT and FIELD statements) for inclusion as members of a PDS source library
- Data base segment rules source (IMSADF II SYSTEM statement, data base SEGMENT and FIELD statements, and GENERATE statement) for creating segment layout and segment handler rules
- Default conversational transaction rule and screen source (IMSADF II SYSTEM statement, data base SEGMENT and FIELD statements, and conversational GENERATE statements), one per data base segment

Also, an installation using the IMSADF II master rule approach can use the ADFIN processor to add and/or update the IMSADF II extensibility subjects in the Data Dictionary.

The combination of the ADFIN and ADFOUT processors can be used to ensure data integrity between the IMSADF II Rules and the Data Dictionary.

PURPOSE

The Data Dictionary Extension easily blends into the IMSADF II application development environment, eliminating redundancy and errors in data definition, and takes advantage of the improved control and documentation that results from using the DB/DC Data Dictionary.

The IMSADF II Data Dictionary Extension:

- Extends Data Dictionary enforced standards over IMSADF II applications.
- Provides consistent views of the data base to all applications (IMSADF II, non-IMSADF II (PL/I, COBOL), Special Processing Routines, etc.).
- Generates default, single segment IMSADF II transactions to display and modify data base segments defined in the Data Dictionary.
- Generates Data Dictionary commands to update the Dictionary with information from existing, new, and/or changing IMSADF II applications.

- Extends the standard types of Dictionary reports to IMSADF II, giving the user easy access to information, such as WHERE-USED, WHAT-USED, and IMSADF II-PL/I-COBOL data element name cross references.
- Assists in migration from test to production.
- Allows future extensions.

BENEFITS

Users of the Data Dictionary Extension do not have to duplicate IMSADF II segment and field physical descriptions. Without the Data Dictionary Extension they must be defined once to the Data Dictionary and once to the IMSADF II Rules Generator. IMSADF II definitions necessary to support the master rules concept can be entered, deleted, updated and maintained in the Data Dictionary Extension model through the Data Dictionary Extension's ADFIN processor or through the standard Data Dictionary online and batch facilities.

The Data Dictionary Extension ADFOUT processor allows users to extract segment and field information from the Data Dictionary, eliminating definition errors. The Data Dictionary becomes a focal point for controlling the entire application development environment.

USER REQUIREMENTS

Before an installation can begin using the IMSADF II Data Dictionary Extension, the following must be done within the installation's Data Dictionary:

1. Define the application data base(s), that is the segments, the elements, and the relationships between them. In most instances this information should already be in the appropriate standard categories of the Data Dictionary as part of the data base design. Existing applications that have not been defined to the Data Dictionary can be added via the Data Dictionary's structures-in process.

2. Install the IMSADF II Data Dictionary Extension model in the installation Data Dictionary. The installation process is described in Chapter 2, "The Model."

3. Define IMSADF II extensibility subjects. These subjects can be added to the Data Dictionary by using the IMSADF II ADFIN Processor or the Dictionary Interactive Display Forms Facility, or by executing the Data Dictionary online or batch commands. These procedures are discussed in Chapter 2, "The Model."

SPECIAL CONSIDERATIONS

Before the IMSADF II Data Dictionary Extension can be installed, the IBM OS/VS DB/DC Data Dictionary (5740-XXF), Release 5.0 or later must be installed.

The Data Dictionary Program Access Facility (PAF) and Extensibility Facility are used to define and access the IMSADF II Data Dictionary Extension. For additional information on PAF and on the Extensibility Facility and its installation process, refer to the DB/DC Data Dictionary Administration and Customization Guide, SH20-9174.

MASTER RULE CONCEPT

The master rule concept is: a common set of data base rules that can be defined and shared across an entire IMSADF II application. Options specific to individual transactions are coded and generated into their specific transaction rules and screens. Any editing or processing options that are applied to the 'master' data base rules are common for all transactions in the application. By using the master rules approach, all transactions within an application view the data base environment in a consistent manner.

If an installation uses the master rule concept when developing its applications, the IMSADF II Data Dictionary Extension can be used to generate a set of online conversational transactions (one per data base segment) to manipulate any segment in their IMS/VS data base. These transactions can be used during the test stage of the IMSADF II application development cycle for creating test data and verifying results. They also become data base maintenance transactions once the application goes into production.

In addition, they may promote end user involvement very early in the development cycle by providing a convenient means for the end user to participate in test data input and to verify results. These master rule transactions give the end user an early opportunity to become accustomed to the sign-on and menu processing benefits offered by IMSADF II. Early end user involvement with an application is very important for the following reasons:

- Early detection of logic or design errors
- Prototyping and modeling
- Screen design
- Data attributes and editing
- Testing
- Owning the application

By using the master rules concept, the application developer only provides unique end user application functions. Items such as support transactions, consistent description of the data base environment and screen and transaction flow are automatically provided by the master rules and the use of IMSADF II. With these functions, IMSADF II can provide increased programmer productivity.

Master rules are contained within and controlled by the Data Dictionary. Unique end-user transactions are developed outside the control of the Data Dictionary; however, every end-user transaction is a merge of common master rule source and unique application source.

MASTER RULE SUPPORT

The ADFIN Extract and Format and ADFOUT processors support the Master Rule environment.

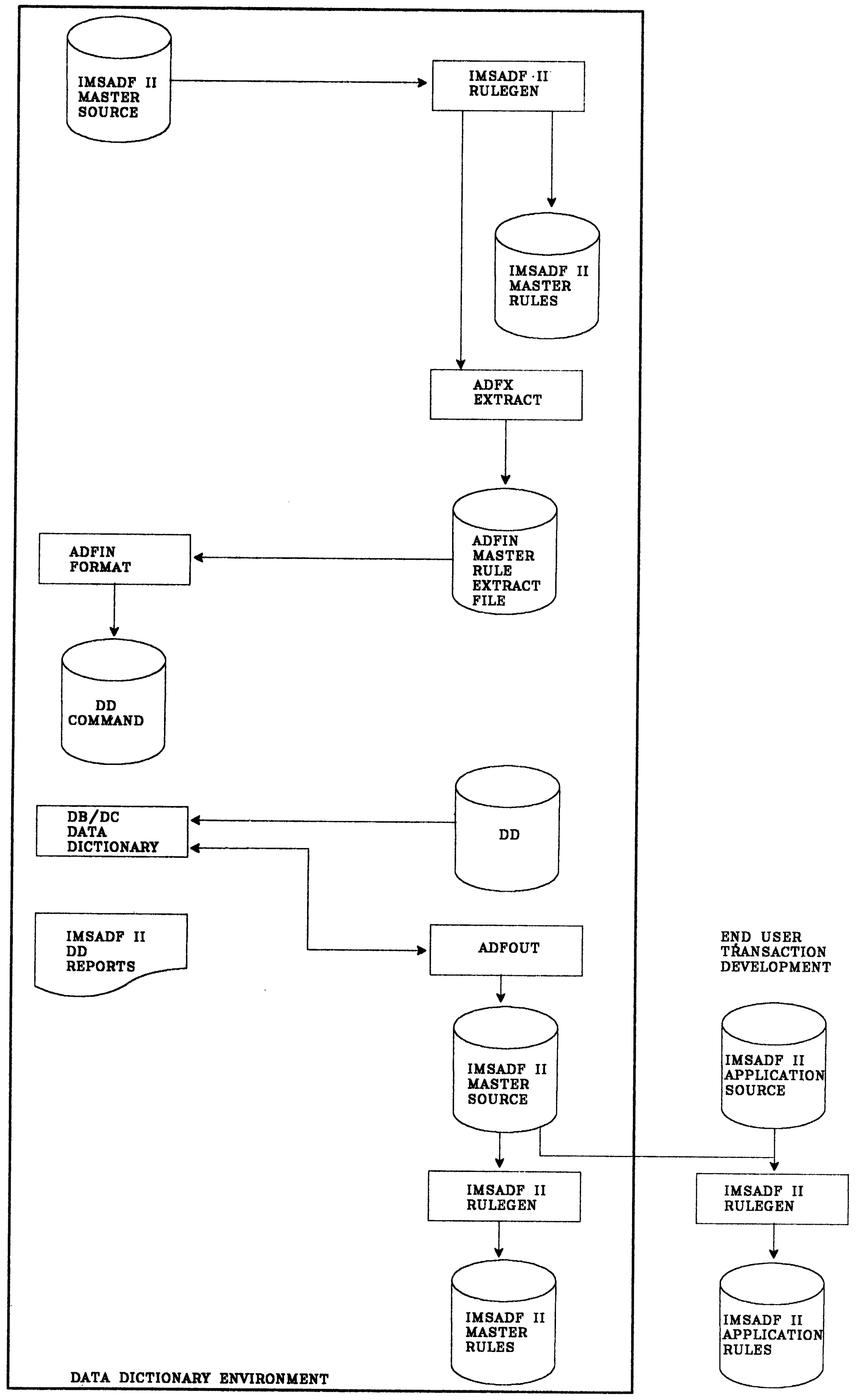

Figure 1-1. IMSADF II Master Rule - Data Dictionary Environment

USING THE IMSADF II DATA DICTIONARY EXTENSION

Information must be entered into the Data Dictionary using standard Data Dictionary commands, such as COBOL_IN, PLI_IN, DBD_IN, ADD and ADD_RELATIONSHIP, the Dictionary batch forms facility, the Interactive Display Form Facility, or by using the IMSADF II ADFIN processor.

You can then invoke the IMSADF II Data Dictionary Extension ADFOUT processor with the Dictionary EXECUTE command. The output from this processor consists of source statements for the Rules Generator. This output can be directed to a library and used in conjunction with Rules Generator INCLUDE statements or it can be processed directly by the Rules Generator. Figure 1-2 illustrates these choices.

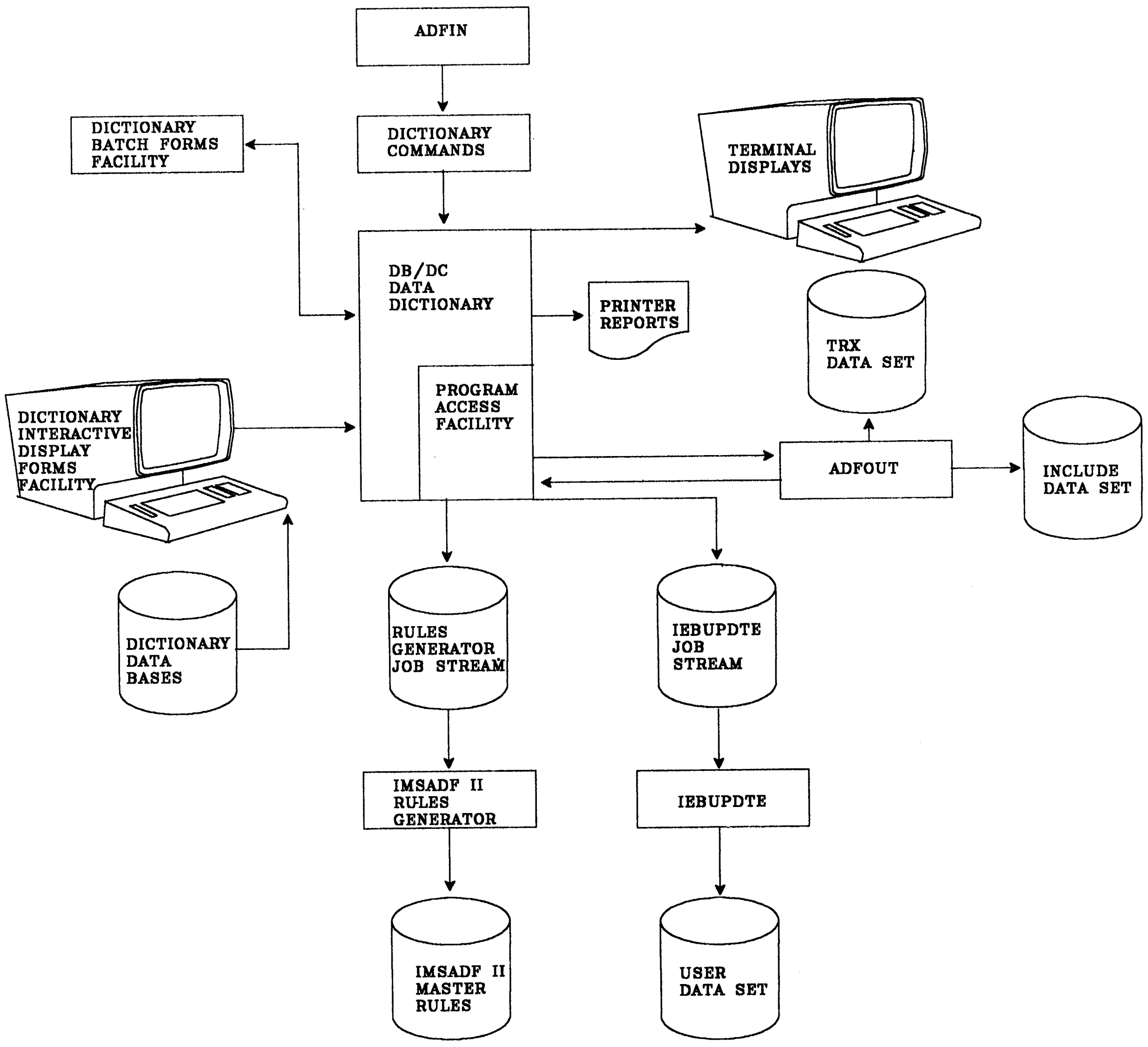

Figure 1-2. Using the IMSADF II Data Dictionary Extension

CHAPTER 2. THE MODEL

The IMSADF II Data Dictionary Extension model is a defined layout for additional Rules Generator operands to be added to the DB/DC Data Dictionary. The model is defined and built using the DB/DC Data Dictionary Extensibility Facility. This facility allows additional resources to be handled that are not defined by the standard categories of the Data Dictionary.

Three extensibility categories and five extensibility relationships are defined in the IMSADF II Data Dictionary Extension model. The model also uses five standard DB/DC Data Dictionary categories: PSB, PCB, DATABASE, SEGMENT and ELEMENT, and their corresponding relationships. The layout of the model is shown in Figure 2-1.

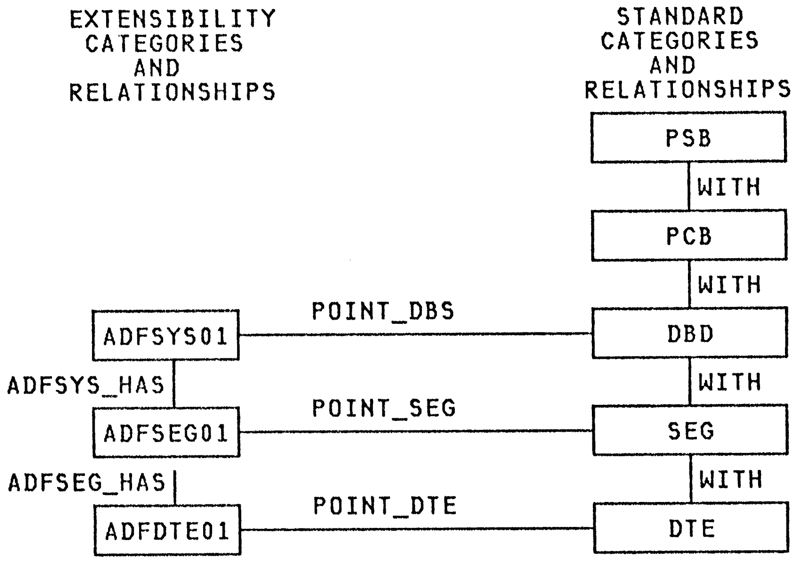

Figure 2-1. IMSADF II Data Dictionary Extension Model

EXTENSIBILITY FACILITY

The DB/DC Data Dictionary Extensibility Facility permits an installation to extend its Data Dictionary 'logically'. It allows new categories to be defined in an installation's Data Dictionary. These categories may represent any entity of importance to an installation's data processing environment.

A category is the basic structural element of the Data Dictionary. It provides support for entries of specific subjects. A subject entry is created in a particular category by defining a name for it. With the Extensibility facility, each installation can define up to 200 new categories.

Similarly, an installation can define new types of relationships. One or more different types of relationships can be defined between installation-defined categories or between an installation-defined category and a standard category.

A relationship is a logical connection that exists between two subjects. Relationships are created to show a dependency among categories. With the Extensibility facility, each installation can define an unlimited number of relationships as long as at least one extensibility category is involved.

For additional information on the Extensibility Facility and its installation process, refer to the DB/DC Data Dictionary Administration and Customization Guide, SH20-9174.

ADDITIONAL EXTENSIBILITY CATEGORIES

ADFSYS01 CATEGORY

The ADFSYS01 category records information necessary to generate a common IMSADF II SYSTEM statement definition. This category has a relationship to and from the standard DATABASE category. The user name portion of the ADFSYS01 category subject-name consists of the four-character IMSADF II application system ID (SYSID). The attributes associated with the ADFSYS01 category are shown in Figure 2-2.

ATTRIBUTE	LENGTH	DESCRIPTION
SOMTX	2	IMS/VS transaction cluster code
PGROUP	2	project group ID
AGROUP	4	audit group ID
PCBNO	1-3	PCB number in PSB
DBID	2	data base ID
ADFID	4	Application Development Facility ID
STRAILER	1	screen name trailer
SHEADING	1-54	screen heading
SFORMAT	4-5	screen format for literals
MAXKEY	1-3	screen key length

Figure 2-2. ADFSYS01 Category Attributes

All ADFSYS01 attributes are optional. An ADFSYS01 subject must be defined and must be related to the standard DATABASE subject.

If an installation is going to use ADFOUT to generate default master rule transactions directly with the Rules Generator source derived from the Data Dictionary Extension model, the first two attributes, SOMTX and PGROUP, are required by the Rules Generator. If default master rule transactions are going to be created, it is recommended that the SHEADING attribute also be entered. This makes the default screens more user friendly.

The first two attributes, SOMTX and PGROUP, can also be passed to the ADFOUT processor as input parameters. If they are passed as input parameters, they override their corresponding attribute value.

If an attribute is left blank, the corresponding keyword is not created by ADFOUT. If the output is passed to the Rules Generator, defaults are taken for those attributes not defined. If a value has been entered for an attribute, it is generated every time the SYSTEM statement output is created.

If the ADFIN processor is to be used to create Data Dictionary commands to add or update the ADFSYS01 category, refer to the _IMS Application Development Facility II Version 2 Release 2 Application Development Reference_ for further information regarding the Rules Generator SYSTEM statement. The SYSID keyword is required by the Rules Generator and is used by ADFIN for the user name of the ADFSYS01 subject. The SOMTX keyword is also required by the Rules Generator. Its value will be used to update the SOMTX attribute of the ADFSYS01 subject.

Although the PGROUP keyword is not required by the Rules Generator, it is necessary for installations that plan to use the ADFOUT processor to generate default master rule transactions. This keyword is used to update the PGROUP attribute of the ADFSYS01 subject, which is required by ADFOUT. All other IMSADF II Rules Generator SYSTEM statement keywords have default values as described in the _IMS Application Development Facility II Version 2 Release 2 Application Development Reference_. If a keyword is not coded on the SYSTEM statement, its default value will be used to update the corresponding attribute of the ADFSYS01 subject.

ADFSEG01 CATEGORY

The ADFSEG01 category records the IMSADF II segment IDs and associated attributes related to each data base standard segment. This category has a relationship to and from the standard SEGMENT category. The user name portion of the ADFSEG01 category subject-name consists of the four-character application system ID and the two-character IMSADF II segment ID. The attributes associated with the ADFSEG01 category are shown in Figure 2-3.

The application system ID is present in the user name to allow installations to define unique ADFSEG01 subject-names. Its presence allows multiple IMSADF II segments, with the same IMSADF II segment ID, to be related to a single standard segment; this standard segment is related to multiple data bases that are defined to multiple application systems.

The IMSADF II segment ID is also used to define parentage in other IMSADF II segment definitions, Rules Generator operand PARENT=SEGID. If multiple IMSADF II segment definitions are related to the same standard segment then the application segment ID from the first valid IMSADF II segment found is used to define parentage.

ATTRIBUTE	LENGTH	DESCRIPTION
PCBNO	1-3	PCB number in PSB
DBID	2	data base ID
TRAILER	2	trailer for data base name
DCFIELD	1-4	field to validate prior to update
SKLEFT	1-36	header for secondary key selection screen - occurs twice
SKRIGHT	1-36	header for secondary key selection screen - occurs twice
SKSEGS	1-2	# of segments for secondary key selection screen
KASCEND	1-3	key values ascend or are unordered
ADBSNAME	1-8	data base name
AKEYNAME	1-8	DBD sequence field name
ALENGTH	1-8	segment length

Figure 2-3. ADFSEG01 Category Attributes

Note: The last three attributes in Figure 2-3 are not standard Rules Generator segment operands and are only required by ADFOUT under special circumstances.

All ADFSEG01 attributes are optional. An ADFSEG01 subject must be defined and must be related to a standard segment even if no attributes are specified for the ADFSEG01 subject.

If an attribute is left blank, the corresponding keyword is not created by ADFOUT. If the output is passed to the Rules Generator, defaults are taken for those attributes not defined. If a value has been entered for an attribute, it is used every time the SEGMENT statement output is created.

The ADBSNAME attribute is required when a standard segment is related to multiple data bases within a single application system (e.g., logical and physical DBD). This requires that a unique IMSADF II segment ID be defined in the ADFSEG01 category for each data base view. There would be multiple IMSADF II segment definitions, having the same application system ID but different IMSADF II segment IDs, related to the same standard segment. This is still not enough information for the ADFOUT processor to determine which segment definition to select; therefore, the ADBSNAME attribute is used to determine which IMSADF II segment ID is correct for a given data base view. When used, the ADBSNAME attribute contains the one- to eight-character data base name.

The ADFOUT processor always verifies this attribute. If the ADBSNAME attribute is blank, the current IMSADF II segment is processed without comparing any other segments. If the attribute is not blank, it is compared with the user name portion of the current data base being processed. If the names match, the current IMSADF II segment is processed. If the names do not match, the current IMSADF II segment is not processed and any additional segments are analyzed.

In order for any IMSADF II segment definition in the ADFSEG01 category to be processed by ADFOUT, the following criteria must be met:

- IMSADF II segment ID must be defined in the input parameter string, either by SEG=ALL or SEG=(list of segment IDs).
- If not directly specified in the list of segment IDs, the segment ID can be inferred if it is a parent segment of one of the list of segment IDs.
- The system ID portion of the ADFSEG01 subject-name must match the current application system ID being processed.
- The ADBSNAME attribute must be blank or match the current data base name being processed.

Therefore, care must be taken when relating multiple IMSADF II segments to a single standard segment. The ADBSNAME attribute should be entered for all IMSADF II segments when multiple IMSADF II segments are related to a single standard segment. If this is not done and the first IMSADF II segment definition found has a blank ADBSNAME attribute, it is used by ADFOUT even if a subsequent IMSADF II segment definition has a valid ADBSNAME attribute.

Multiple IMSADF II segment definitions related to the same standard segment also allow the user to enter IMSADF II alias segment definitions. These definitions are essentially identical and can be used in conjunction with IMSADF II twin processing transactions.

In most cases, the AKEYNAME attribute can be left blank. It is only required when the DBD sequence field for a segment is not contained within its field definitions or when another field is to be used as the key (e.g. for secondary indexing). If the AKEYNAME attribute is blank, the IMSADF II segment KEYNAME parameter is derived from the assembler name of the DBD sequence field of the current standard segment. If the AKEYNAME attribute is not blank, it is used.

In most cases, the ALENGTH attribute can be left blank. It is required by ADFOUT only when the IMSADF II segment defines a segment that is processed by a secondary index, where the search field is not a part of the segment. This implies that the length of the segment defined to IMSADF II is greater than the true segment length. If the ALENGTH attribute is not blank, it must be equal to or greater than the standard segment MAXBYTES attribute. If the ALENGTH attribute is less then MAXBYTES, it is ignored by ADFOUT.

If the ADFIN processor is to be used to create Data Dictionary commands to add and/or update the ADFSEG01 category, refer to the _IMS Application Development Facility II Version 2 Release 2 Application Development Reference_ for further information regarding the Rules Generator SEGMENT statement. The ID keyword is required by the Rules Generator and is concatenated with the SYSID to be used by ADFIN for the user name of the ADFSEG01 subject.

The PCBNO keyword is only required by the Rules Generator when it differs from the PCBNO on the SYSTEM statement. Otherwise, its default value is the same as the PCBNO on the SYSTEM statement. The PCBNO keyword value or its default will be used by ADFIN to update the PCBNO attribute of the ADFSEG01 subject.

The DCFIELD keyword is not required by the Rules Generator, but when it is coded, it will be used by ADFIN to update the DCFIELD attribute of the ADFSEG01 subject. When the DCFIELD keyword is not coded and the DCFIELD attribute of the ADFSEG01 subject is non-blank, ADFIN will create a Dictionary command to blank out this attribute. This is also true for the SKLEFT and SKRIGHT attributes.

The ADBSNAME attribute is updated using the one- to eight-character username of the DATABASE subject related to the PCB subject which corresponds to the segment's PCBNO attribute. The KEYNAME keyword is not required by the Rules Generator, but when it is coded, it will be used by ADFIN to update the AKEYNAME attribute of the ADFSEG01 subject. When the KEYNAME keyword is not coded and the AKEYNAME attribute of the ADFSEG01 subject is non-blank, ADFIN will create a Dictionary command to blank out this attribute.

The LENGTH or BYTES keyword is required by the Rules Generator and its value will be used to update the ALENGTH attribute of the ADFSEG01 subject. All other IMSADF II Rules Generator SEGMENT statement keywords have default values as described in the IMS Application Development Facility II Version 2 Release 2 Application Development Reference. If a keyword is not coded on the SEGMENT statement, its default value will be used to update the corresponding attribute of the ADFSEG01 subject.

ADFDTE01 CATEGORY

The ADFDTE01 category records the IMSADF II field IDs and associated attributes related to a standard data element. This category has pointers to and from the standard ELEMENT category.

While all ADFDTE01 attributes are optional, an ADFDTE01 subject must be defined for and related to every standard ELEMENT that is to be processed by ADFOUT. When a CONTAINS relationship exists, this includes:

- standard ELEMENTs which contain other ELEMENTs,
- the contained standard ELEMENTs.

ADFOUT will not process a contained ELEMENT whose containing ELEMENT does not have an associated ADFDTE01.

Note: An additional requirement is that all standard ELEMENTs be directly related to the standard SEGMENT that contains them. If lower level elements are related only to the containing ELEMENT, ADFOUT will not process them. Use of the Dictionary EXTEND_RELATIONSHIP (XR) command is recommended when SEGMENT to ELEMENT relationships must be added for contained ELEMENTs.

The user name portion of the ADFDTE01 category subject-name consists of three parts:

1. An optional two-character IMSADF II application system ID,
2. An optional two-character IMSADF II segment ID,
3. And a required one- to four-character IMSADF II field ID.

The attributes associated with the ADFDTE01 category are shown in Figure 2-4.

To identify an IMSADF II field, unique to an IMSADF II segment and/or unique to an IMSADF II system, the user name portion of the ADFDTE01 category subject-name can be built in several ways:

- If there is a one-to-one relationship between the standard element and the IMSADF II element, only the one- to four-character IMSADF II field ID must be entered. The additional qualifiers are optional for this case.
- If multiple IMSADF II elements are related to a single standard element, two approaches can be used to uniquely identify each IMSADF II element:
  1. Add the first two characters of the IMSADF II system ID and the two-character IMSADF II segment ID before the one- to four-character IMSADF II field ID. This uniquely relates this IMSADF II field definition to a single IMSADF II segment and system.

2. Define the one- to four-character IMSADF II field ID in the user name portion of the ADFDTE01 category subject-name to carry the additional IMSADF II field-ID qualifiers in two optional ADFDTE01 attributes. The SYSID attribute is used to define the four-character IMSADF II system ID, and the SEGID attribute is used to define the two-character IMSADF II segment ID.

For all cases, if the non-blank portion of the user name of the ADFDTE01 category subject-name exceeds four characters, the first four characters are assumed to be the two-character IMSADF II system ID and the two-character IMSADF II segment ID. The remaining four characters are taken to be the actual IMSADF II field ID.

Note: When relating multiple IMSADF II fields to a single standard element, the IMSADF II system ID and segment ID should be specified either as part of the user name or in the optional IMSADF II field attributes. If this is not done and the first IMSADF II field definition found has only the field ID specified, it is used. However, a subsequent IMSADF II field definition may be the one actually required.

ATTRIBUTE	LENGTH	DESCRIPTION
KEY	1-3	key field
SIGN	1-3	decimal or packed decimal field, plus or minus, an indicator in stored value
SNAME	1-30	screen literal for field
RELATED	1-3	request for display on Secondary Key selection screen
COLUMN	1-3	field column position on Secondary Key selection screen
AUDIT	1-3	request for field audit
CAUDIT	1-3	request for common field audit
MSG	1-3	request for message sending
SYSID	4	optional IMSADF II system ID
SEGID	2	optional IMSADF II segment ID
OTYPE	1-5	Request for override of standard data element type attribute

Figure 2-4. ADFDTE01 Category Attributes

All ADFDTE01 attributes are optional.

If an attribute is left blank, the corresponding keyword is not created by ADFOUT. If the output is passed to the Rules Generator, defaults are taken for those attributes not defined. If a value is entered for an attribute, it is used every time the FIELD statement is created. The IMSADF II field TYPE operand is derived from the ELEMENT TYPE attribute, which does not have an equivalent to NUM, DATE, DBCS, or MIXED. ADFDTE01 OTYPE overrides ELEMENT TYPE to produce TYPE=NUM, TYPE=DATE, TYPE=DBCS, or TYPE=MIXED on the IMSADF II FIELD statement.

SYSID and SEGID are not standard Rules Generator FIELD statement parameters and are required only under special circumstances. SYSID and SEGID should be left blank except where they are needed to relate an IMSADF II field to a unique IMSADF II segment and/or IMSADF II system.

If an installation is going to generate default master rule transactions directly, using the Rules Generator source derived from the Data Dictionary Extension model, at least one IMSADF II field per IMSADF II segment must contain the KEY=YES parameter. If none have KEY=YES, the Rules Generator terminates with an error message, unless a field override statement has been added to the Rules Generator source prior to invoking the Rules Generator. Also, it is recommended that the SNAME attribute be entered for each field to make default screens more meaningful.

If the ADFIN processor is to be used to create Data Dictionary commands to add and/or update the ADFDTE01 category, refer to the IMS Application Development Facility II Version 2 Release 2 Application Development Reference for further information regarding the Rules Generator FIELD statement. The ID keyword is required by the Rules Generator. ADFIN will first determine if the ADFDTE01 subject with the concatenated user name (two-character SYSID concatenated with two-character segment ID concatenated with four-character field ID) exists. If so, this ADFDTE01 subject will be updated by ADFIN. If the ADFDTE01 subject with the concatenated user name is not present in the Data Dictionary, ADFIN will determine if the ADFDTE01 subject with a four-character field ID user name exists. If this subject's SYSID and SEGID attributes have values which are the same as the SYSID and SEGID from the Rules Generator source, or if these attributes are blank, this ADFDTE01 subject will be updated by ADFIN. If neither of these ADFDTE01 subjects exist, ADFIN will create a batch command to add an ADFDTE01 subject with the concatenated user name. Under no circumstances will ADFIN update the SYSID or SEGID attributes of any ADFDTE01 subject.

The OTYPE attribute value is derived from the value of the Rules Generator TYPE keyword value. The Data Dictionary standard ELEMENT TYPE attribute does not have an equivalent to the IMSADF II Rules Generator field TYPE keyword value of NUM, ALPHA, DATE, DBCS, or MIXED. The ADFDTE01 OTYPE attribute is updated by ADFIN to override the standard ELEMENT TYPE attribute when the Rules Generator field TYPE keyword has a value of NUM, ALPHA, DATE, DBCS, or MIXED. When the IMSADF II Rules Generator field TYPE keyword has a value which does have an equivalent in the Data Dictionary standard ELEMENT TYPE attribute, and the ADFDTE01 OTYPE attribute is non-blank, ADFIN will create a Dictionary command to blank out this attribute.

All other IMSADF II Rules Generator FIELD statement keywords have default values as described in the IMS Application Development Facility II Version 2 Release 2 Application Development Reference. If a keyword is not coded on the FIELD statement, its default value will be used to update the corresponding attribute of the ADFDTE01 subject.

See the IMS Application Development Facility II Version 2 Release 2 Application Development Reference for additional information on the Rules Generator operands supported by the Data Dictionary Extension model.

ADDITIONAL EXTENSIBILITY RELATIONSHIPS

The additional extensibility relationships that are defined in the Data Dictionary Extension model relate a standard category to a new extensibility category and relate the extensibility categories to each other. There are no attributes associated with these relationships. None of the relationships are sequenced or directed.

ADFSYS01/TO/DBS RELATIONSHIP

The ADFSYS01/TO/DBS relationship defines the data bases which are part of the IMSADF II systems data base environment and determines their relative sequencing within the system. The forward relationship keyword is POINT_DBS. The inverse relationship keyword is POINT_ADFSYS.

ADFSEG01/TO/SEG RELATIONSHIP

The ADFSEG01/TO/SEG relationship relates the standard SEGMENT category to the ADFSEG01 category. The forward relationship keyword is POINT_SEG. The inverse relationship keyword is POINT_ADFSEG.

ADFDTE01/TO/DTE RELATIONSHIP

The ADFDTE01/TO/DTE relationship relates the standard ELEMENT category to the ADFDTE01 category. The forward relationship keyword is POINT_DTE. The inverse relationship keyword is POINT_ADFDTE.

ADFSYS01/TO/ADFSEG01 RELATIONSHIP

The ADFSYS01/TO/ADFSEG01 relationship relates the ADFSYS01 category to the ADFSEG01 category. The forward relationship is ADFSYS_HAS. This is an optional relationship; however, it is updated by the ADFIN processor.

ADFSEG01/TO/ADFDTE01 RELATIONSHIP

The ADFSEG01/TO/ADFDTE01 Relationship relates the ADFSEG01 category to the ADFDTE01 category. The forward relationship is ADFSEG_HAS. This is an optional relationship; however, it is updated by the ADFIN processor.

DEFINING THE MODEL

Two steps are involved in defining the IMSADF II Data Dictionary Extension model:

1. Definition process

   Definition is providing extensibility control information (ECI) about what data is acceptable for the subjects belonging to these new extensibility categories and relationships.

2. Installation process

   Installation is making this extensibility control information operational; it is required before user subject data can be entered into the Data Dictionary Extension model.

META DEFINITION

The meta definition process is the establishment of new extensibility categories and relationships in the Data Dictionary. The meta definition for a category or relationship establishes a set of attributes or properties defining the subjects that installations will enter into these new extensibility categories and relationships.

The extensibility control information required to define the additional categories and relationships for the Data Dictionary Extension model are stored in three special standard Data Dictionary categories: CATEGORY, RELTYPE and ATTRTYPE. These categories reside in the extensibility data base.

The CATEGORY category contains the definition of each of the new subject categories: ADFSYS01, ADFSEG01 and ADFDTE01. Each definition contains the subject category's characteristics and naming conventions.

The RELTYPE category contains relationship definitions for the new relationships: ADFSYS01/TO/DBS, ADFSEG01/TO/SEG, ADFDTE01/TO/DTE, ADFSYS01/TO/ADFSEG01, and ADFSEG01/TO/ADFDTE01. Each definition identifies the pair of subject categories participating in the relationship, and specifies the forward and inverse relationship keywords.

The ATTRTYPE category contains the definitions for category and relationship attributes. The same attribute definition can be related to one or more subject categories and relationship definitions. An attribute expresses some property of a category or relationship. The definition of an attribute contains three parts:

- Its properties - a description of what the attribute values represent.

- Its format - how values for the attribute are specified in commands.

- A validation of the defined values.

The definition of an attribute includes its name, keyword and rules for verifying input values for the attribute. The KEYWORD parameter specifies the keyword that is to be used to reference the attribute in commands and display forms.

DEFINITION PROCESS

The control information defining the meta definition of the Data Dictionary Extension model is contained in the IMSADF II supplied library, IMSADF.JCLLIB member DDEECI. The DDEECI member contains batch Data Dictionary commands that must be processed using the standard Data Dictionary batch processor JCL. See the IMS Application Development Facility II Version 2 Release 2 Installation Guide for information on the library.

Every extensibility category that is defined to the Data Dictionary must be assigned a unique subject code number. This number is used internally by the Data Dictionary to distinguish an extensibility category from any other category. The valid range for extensibility category subject code numbers is 56-255. The default subject code numbers assigned to the three extensibility categories that make up the Data Dictionary Extension model are: ADFSYS01 - 100, ADFSEG01 - 101 and ADFDTE01 - 102. These extensibility category subject code numbers can be modified to meet each installation's requirements by editing the DDEECI member prior to executing the Data Dictionary batch processor. These values must be coded on SBJCODE keyword on the three ADD CATEGORY batch commands.

Before executing the DDEECI member, you may also need to invoke the SIGN_ON command. A SIGN_ON command is included as a comment at the beginning of the DDEECI member. Remove the * from column 1 and enter a valid user ID and password if your installation uses Data Dictionary security.

INSTALLATION PROCESS

Before an installation can begin defining subjects in the Data Dictionary Extension model, the model's extensibility control information (ECI) must be put into the Data Dictionary. This installation process is accomplished with the Data Dictionary INSTALL command. Installation causes the establishment of extensibility control information so that the meta definitions become operational. Once installed, a category or relationship definition cannot be modified.

Installation of the Data Dictionary Extension model, except for subject code number, is standard for all users. The model's meta definition should not be modified prior to installation. The Data Dictionary INSTALL commands, that are used to install the Data Dictionary Extension model, are also contained in member DDEECI.

For new users of the Data Dictionary Extension, the definition and installation process for the Data Dictionary Extension model are completed with a single execution of the Data Dictionary batch processor using the IMSADF.JCLLIB library, DDEECI member as the DDINPUT. Those installations which installed the IMSADF II Version 1, Release 1 Data Dictionary Extension on Release 5 of the Data Dictionary may modify the Version 1, Release 1 model by executing the Data Dictionary batch library DDEECI5 member as DDINPUT.

Figure 2-5 through Figure 2-12 contain examples of the Data Dictionary Guide reports that represent the Data Dictionary Extension model after it has been successfully installed. The commands to generate these Guide reports are included in member DDEECI and are part of the output generated by the installation process for the Data Dictionary Extension model. These Guide reports provide the Category and Reltype names and a list of subject attributes. You should verify that your reports are consistent with these.

```
 * * * * * * * *         DB/DC DATA DICTIONARY REPORT     09/13/84  14:51:49
                               CATEGORY GUIDE              PAGE:  01

CATEGORY:  ADFSYS01

SUBJECT NAME RULES:                        DATE:
  NAME TYPE:     N
  MIN LENGTH:    04
  MAX LENGTH:    04
  VAL ROUTINE:

ATTRIBUTES:

         REPEAT   LENGTH      DATA   VALIDATION
KEYWORD  FACTOR  MIN    MAX   TYPE   TYPE  VALUES
-------  ------  ---    ---   ----   ----  ------

SOMTX             2      2     C     NONE
PGROUP            2      2     C     NONE
AGROUP            4      4     C     NONE
PCBNO             1      3     Z     RNG   001,120
DBID              2      2     C     NONE
ADFID             4      4     N     NONE
STRAILER          1      1     C     NONE
SHEADING          3     56     Q     NONE
SFORMAT           4      5     C     LIST  DASH ,LEFT ,RIGHT
MAXKEY            1      3     Z     RNG   001,100
  * * * * * * * *         DB/DC DATA DICTIONARY REPORT     09/13/84  14:51:49
                               CATEGORY GUIDE              PAGE:  02


RELATIONSHIP TYPES:

FORWARD NAME      TO CATEGORY     INVERSE NAME    SEQOPT     SEQATTR     DIRECTED
------------      -----------     ------------    ------     -------     --------
ADFSYS_HAS        ADFSEG01        IN_ADFSYS         N                    NONE
POINT_DBS         DATABASE        POINT_ADFSYS      N                    NONE
                      * * *  END-OF-REPORT  * * *
```

Figure 2-5. GUIDE Report for ADFSYS01 Category

```
 * * * * * * * *        DB/DC DATA DICTIONARY REPORT      09/13/84  14:51:54
                              RELTYPE  GUIDE               PAGE:  01

RELTYPE:   ADFSYS01/TO/DBS

RELATIONSHIP TYPE RULES:                       DATE:
  LEFT-HAND CATEGORY:   ADFSYS01
  RIGHT-HAND CATEGORY:  DATABASE
  FORWARD NAME:         POINT_DBS
  INVERSE NAME:         POINT_ADFSYS
  SEQUENCE OPTION:      NO
  SEQUENCE ATTRIBUTE:
  DIRECTED:             NO

ATTRIBUTES:

         REPEAT   LENGTH     DATA   VALIDATION
KEYWORD  FACTOR  MIN   MAX   TYPE  TYPE  VALUES
-------  ------  ---   ---   ----  ----  ------

                   * * *  END-OF-REPORT  * * *
```

Figure 2-6. GUIDE Report for ADFSYS01/TO/DBS Relationship-Type

```
 * * * * * * * *        DB/DC DATA DICTIONARY REPORT      09/13/84  09:08:09
                              RELTYPE  GUIDE               PAGE:  01

RELTYPE:   ADFSYS01/TO/ADFSEG01

RELATIONSHIP TYPE RULES:                       DATE:
  LEFT-HAND CATEGORY:   ADFSYS01
  RIGHT-HAND CATEGORY:  ADFSEG01
  FORWARD NAME:         ADFSYS_HAS
  INVERSE NAME:         IN_ADFSYS
  SEQUENCE OPTION:      NO
  SEQUENCE ATTRIBUTE:
  DIRECTED:             NO

ATTRIBUTES:

         REPEAT   LENGTH     DATA   VALIDATION
KEYWORD  FACTOR  MIN   MAX   TYPE  TYPE  VALUES
-------  ------  ---   ---   ----  ----  ------

                   * * *  END-OF-REPORT  * * *
```

Figure 2-7. GUIDE Report for ADFSYS01/TO/ADFSEG01 Relationship-Type

```
 * * * * * * * *          DB/DC DATA DICTIONARY REPORT     09/13/84  14:51:50
                                CATEGORY GUIDE              PAGE:  01

CATEGORY:  ADFSEG01

SUBJECT NAME RULES:                              DATE:
  NAME TYPE:     N
  MIN LENGTH:    06
  MAX LENGTH:    06
  VAL ROUTINE:

ATTRIBUTES:

         REPEAT   LENGTH      DATA   VALIDATION
KEYWORD  FACTOR  MIN   MAX    TYPE   TYPE  VALUES
-------  ------  ---   ---    ----   ----  ------

ADBSNAME          1     8      C     NONE
PCBNO             1     3      Z     RNG   001,120
DBID              2     2      C     NONE
TRAILER           2     2      C     NONE
DCFIELD           1     4      C     NONE
SKLEFT    02      3    38      Q     NONE
SKRIGHT   02      3    38      Q     NONE
SKSEGS            1     2      Z     RNG   00,38
KASCEND           1     3      C     LIST  YES,NO ,Y  ,N
AKEYNAME          1     8      C     NONE
ALENGTH           1     8      Z     NONE

 * * * * * * * *          DB/DC DATA DICTIONARY REPORT     09/13/84  14:51:50
                                CATEGORY GUIDE              PAGE:  02

RELATIONSHIP TYPES:

FORWARD NAME     TO CATEGORY     INVERSE NAME    SEQOPT     SEQATTR     DIRECTED
------------     -----------     ------------    ------     -------     --------
ADFSEG_HAS       ADFDTE01        IN_ADFSEG          N                     NONE
IN_ADFSYS        ADFSYS01        ADFSYS_HAS         N                     NONE
POINT_SEG        SEGMENT         POINT_ADFSEG       N                     NONE
                          * * *  END-OF-REPORT  * * *
```

Figure 2-8. GUIDE Report for ADFSEG01 Category

```
 * * * * * * * *          DB/DC DATA DICTIONARY REPORT     09/13/84  14:51:54
                                RELTYPE  GUIDE              PAGE:  01

RELTYPE:   ADFSEG01/TO/SEG

RELATIONSHIP TYPE RULES:                         DATE:
  LEFT-HAND CATEGORY:  ADFSEG01
  RIGHT-HAND CATEGORY: SEGMENT
  FORWARD NAME:        POINT_SEG
  INVERSE NAME:        POINT_ADFSEG
  SEQUENCE OPTION:     NO
  SEQUENCE ATTRIBUTE:
  DIRECTED:            NO

ATTRIBUTES:

         REPEAT   LENGTH      DATA   VALIDATION
KEYWORD  FACTOR  MIN   MAX    TYPE   TYPE  VALUES
-------  ------  ---   ---    ----   ----  ------

                          * * *  END-OF-REPORT  * * *
```

Figure 2-9. GUIDE Report for ADFSEG01/TO/SEG Relationship-Type

```
 * * * * * * * *         DB/DC DATA DICTIONARY REPORT      09/13/84  09:08:10
                               RELTYPE  GUIDE              PAGE:  01

RELTYPE:   ADFSEG01/TO/ADFDTE01

RELATIONSHIP TYPE RULES:                          DATE:
  LEFT-HAND CATEGORY:   ADFSEG01
  RIGHT-HAND CATEGORY:  ADFDTE01
  FORWARD NAME:         ADFSEG_HAS
  INVERSE NAME:         IN_ADFSEG
  SEQUENCE OPTION:      NO
  SEQUENCE ATTRIBUTE:
  DIRECTED:             NO

ATTRIBUTES:

         REPEAT   LENGTH      DATA   VALIDATION
KEYWORD  FACTOR  MIN   MAX    TYPE  TYPE  VALUES
-------  ------  ---   ---    ----  ----  ------

                    * * *  END-OF-REPORT  * * *
```

Figure 2-10. GUIDE Report for ADFSEG01/TO/ADFDTE01 Relationship-Type

```
 * * * * * * * *         DB/DC DATA DICTIONARY REPORT      09/13/84  14:51:52
                               CATEGORY GUIDE              PAGE:  01

CATEGORY:  ADFDTE01

SUBJECT NAME RULES:                               DATE:
  NAME TYPE:    C
  MIN LENGTH:   01
  MAX LENGTH:   08
  VAL ROUTINE:

ATTRIBUTES:

         REPEAT   LENGTH      DATA   VALIDATION
KEYWORD  FACTOR  MIN   MAX    TYPE  TYPE  VALUES
-------  ------  ---   ---    ----  ----  ------

KEY               1     3      C    LIST  YES,NO ,Y   ,N
SIGN              1     3      C    LIST  YES,NO ,Y   ,N
SNAME             3    32      Q    NONE
RELATED           1     3      C    LIST  YES,NO ,Y   ,N
COLUMN            1     2      Z    RNG   1 ,72
AUDIT             1     3      C    LIST  YES,NO ,Y   ,N
CAUDIT            1     3      C    LIST  YES,NO ,Y   ,N
MSG               1     3      C    LIST  YES,NO ,Y   ,N
SYSID             4     4      C    NONE
SEGID             2     2      C    NONE
OTYPE             1     5      C    LIST  NUM  ,N     ,DATE ,DA   ,D    ,DBCS ,
                                         MIXED,M      ,ALPHA,A

  * * * * * * * *         DB/DC DATA DICTIONARY REPORT      09/13/84  14:51:52
                               CATEGORY GUIDE              PAGE:  02

RELATIONSHIP TYPES:

FORWARD NAME     TO CATEGORY    INVERSE NAME   SEQOPT    SEQATTR     DIRECTED
------------     -----------    ------------   ------    -------     --------

IN_ADFSEG        ADFSEG01       ADFSEG_HAS        N                   NONE
POINT_DTE        ELEMENT        POINT_ADFDTE      N                   NONE
                      * * *  END-OF-REPORT  * * *
```

Figure 2-11. GUIDE Report for ADFDTE01 Category

```
 * * * * * * * *         DB/DC DATA DICTIONARY REPORT     09/13/84  14:51:54
                               RELTYPE  GUIDE              PAGE:  01

RELTYPE:   ADFDTE01/TO/DTE

RELATIONSHIP TYPE RULES:                       DATE:
  LEFT-HAND CATEGORY:   ADFDTE01
  RIGHT-HAND CATEGORY: ELEMENT
  FORWARD NAME:         POINT_DTE
  INVERSE NAME:         POINT_ADFDTE
  SEQUENCE OPTION:      NO
  SEQUENCE ATTRIBUTE:
  DIRECTED:             NO

ATTRIBUTES:

         REPEAT   LENGTH     DATA   VALIDATION
KEYWORD  FACTOR  MIN   MAX   TYPE   TYPE  VALUES
-------  ------  ---   ---   ----   ----  ------

                      * * *  END-OF-REPORT  * * *
```

Figure 2-12. GUIDE Report for ADFDTE01/TO/DTE Relationship-Type

USING THE MODEL

Once the Data Dictionary Extension model is installed, if you plan to use ADFOUT you must enter the additional IMSADF II system, segment and field attributes into the Data Dictionary using the Data Dictionary Extension model. This IMSADF II information can be entered using the IMSADF II ADFIN Processor, or the Data Dictionary batch commands (ADD and ADD_RELATIONSHIP), or it can be entered online through the Dictionary Interactive Display Forms Facility. Refer to the DB/DC Data Dictionary Terminal User's Guide and Command Reference, for additional information on entering data into the Data Dictionary.

ADFSYS01, ADFSEG01 and ADFDTE01 have no required attributes. However, subjects must be defined in these categories and must be related to subjects in appropriate standard categories for the ADFOUT processor to produce correct output.

Figure 2-13, Figure 2-14 and Figure 2-15 show three Subject Specific detail reports for a subject-name in each of the three extensibility categories of the model. Each subject-name, its attributes and relationships were defined using two Data Dictionary batch commands, ADD and ADD_RELATIONSHIP.

```
 * * * * * * * *          DB/DC DATA DICTIONARY REPORT      09/13/84  14:51:49
                                SUBJECT SPECIFIC             PAGE:  01

DICTIONARY DATA BASE: EXT                   CATEGORY: ADFSYS01

NAME:          T  SAMP 0


ATTRIBUTES:

  SOMTX:          OR                            PGROUP:          ZZ
  AGROUP:                                       PCBNO:           001
  DBID:           PA                            ADFID:
  STRAILER:
  SHEADING:       'S A M P L E   P R O B L E M'
  SFORMAT:        DASH                          MAXKEY:

RELATIONSHIPS:

 ADFSYS_HAS     ADFSEG01    TP SAMPPA   0
 POINT_DBS      DATABASE    TP DI21PART 0

                    * * *  END-OF-REPORT  * * *
```

Figure 2-13. ADFSYS01 Subject Specific Detail Report. This is the report for subject-name (T,,SAMP,000) in the ADFSYS01 extensibility category of the IMSADF II Data Dictionary Extension model.

```
 * * * * * * * *          DB/DC DATA DICTIONARY REPORT      09/13/84  14:51:51
                                SUBJECT SPECIFIC             PAGE:  01

DICTIONARY DATA BASE: EXT                   CATEGORY: ADFSEG01

NAME:          T  SAMPBK 0

DESCRIPTION:

  001 BACKORDER

ATTRIBUTES:

  ADBSNAME:                                     PCBNO:
  DBID:                                         TRAILER:
  DCFIELD:
  SKLEFT(01):
  SKLEFT(02):
  SKRIGHT(01):
  SKRIGHT(02):
  SKSEGS:                                       KASCEND:
  AKEYNAME:                                     ALENGTH:

RELATIONSHIPS:

 ADFSEG_HAS     ADFDTE01    T  BKEY      0
 IN_ADFSYS      ADFSYS01    T  SAMP      0
 POINT_SEG      SEGMENT     TC BACKORDR 0
                    * * *  END-OF-REPORT  * * *
```

Figure 2-14. ADFSEG01 Subject Specific Detail Report. This is the report for subject-name (T,,SAMPBK,000) in the ADFSEG01 extensibility category of the IMSADF II Data Dictionary Extension model.

```
 * * * * * * * *          DB/DC DATA DICTIONARY REPORT      09/13/84  14:51:53
                                SUBJECT SPECIFIC             PAGE:  01

DICTIONARY DATA BASE: EXT                  CATEGORY: ADFDTE01

NAME:          T  BKEY 0


ATTRIBUTES:

  KEY:              YES                      SIGN:
  SNAME:            'BACK KEY'
  RELATED:                                   COLUMN:        1
  AUDIT:                                     CAUDIT:
  MSG:                                       SYSID:
  SEGID:                                     OTYPE:

RELATIONSHIPS:

 IN_ADFSEG       ADFSEG01    T  SAMPBK    0
 POINT_DTE       ELEMENT     TC BACKKEY 0
                        * * *  END-OF-REPORT  * * *
```

Figure 2-15. ADFDTE01 Subject Specific Detail Report. This is the report for subject-name (T,,BKEY,000) in the ADFDTE01 extensibility category of the IMSADF II Data Dictionary Extension model.

The subject specific reports in Figure 2-13, Figure 2-14 and Figure 2-15 were created after data was added to the Data Dictionary using the sample procedure. Refer to "Sample Procedure," that follows, for additional information on entering data into the Data Dictionary Extension model.

SAMPLE PROCEDURE

When the Data Dictionary Extension model has been successfully installed, an optional sample procedure can be executed. This procedure can be used to:

- become familiar with the model
- verify that the model was installed correctly
- see the output generated from the model
- demonstrate one method for entering data into the model.

IMSADF.JCLLIB (DDESAMP) contains this sample procedure, which is an 9-step job.

STEP 1 Executes the standard Dictionary batch procedure to delete all the subjects from the existing SAMP system in preparation for adding the new SAMP system.

Note: If the SAMP system is not currently installed in your Data Dictionary, Step 1 will result with a return code of 8.

STEP 2 Executes the standard Dictionary batch procedure. This step documents the DI21PART data base distributed with IMS/VS through the use of DBD_IN and COBOL_IN. It also uses DBD_IN to document the IMSADF II data bases MFDPAR01, MFDPMS01, MFDPSP01, and MFC1WORK[1] and uses PSB_IN for the PSB SAMPTOR. The DBDs must reside in DBD load libraries referenced by the DDDBDLIB DD card

[1] If the installed ADFID is not MFC1, the DBD names are ????AUDT, ????SIGN, ????MSGS, and ????WORK, where ???? is the four-character ADFID value.

and the PSB must be in a PSB load library referenced by the DDPSBLIB DD card. The COBOL data descriptions used by COBOL_IN are found in IMSADF.ADFMAC (SAMPCOB).

STEP 3 Executes the Rules Generator and produces EXTRACT output for the SAMP system. This EXTRACT output is in file &&DDADFX.

STEP 4 Executes the ADFIN FORMAT program, a PAF program. The procedure referenced is the standard Dictionary batch procedure. Input is in file &&DDADFX and output is in &&DDCMD.

STEP 5 Uses IEBGENER utility to print the output from Step 3. This should be compared with Appendix A, "ADFIN Processor Sample Procedure Output" to verify correctness.

STEP 6 Executes the standard Dictionary batch procedure. The DDINPUT DD card must reference the &&DDCMD data set created in Step 3.

STEP 7 Executes the ADFOUT processor, a PAF program. The procedure referenced is the standard Dictionary batch procedure. Output from this step is routed to DINCLUDE and DADFOUT, which are assigned to the printer. It should be compared with Appendix B, "ADFOUT Processor Sample Procedure Output" to verify correctness. It may also be compared with IMSADF.JCLLIB (RGLSAMP), another IMSADF II-supplied member, to aid in understanding which Rules Generator operands are supported by the Data Dictionary Extension.

STEP 8 Executes the standard Dictionary batch procedure to add the structure type ADFDDE containing the IMSADF II extensibility relationships ADFSYS01/TO/ADFSEG01 and ADFSEG01/TO/ADFDTE01. This structure type will be used in Step 9 to delete the sample subject from the IMSADF II extensibility categories.

STEP 9 Executes the standard Dictionary batch procedure to delete all the subjects from the new SAMP system.

OUTPUT

Once the IMSADF II system, segment and field definitions are entered into the Data Dictionary, a variety of outputs can be generated from the data in the Data Dictionary Extension model:

1. Output from the ADFOUT Processor:

   - IMSADF II INCLUDE members for a PDS source library
   - Rules Generator source for segment rules
   - Rules Generator source for default transaction rules and screens

   This output can appear in the data sets referenced by the DDLIST, DDPUNCH, DINCLUDE, DADFOUT or user-specified DD cards or can appear online to the user who invoked the ADFOUT processor, depending on the parameters specified on the EXECUTE command. Refer to "Output Routing" on page 4-4 for additional details.

2. Standard Data Dictionary reports

   This output may be in data sets or may be online to the user.

3. Interactive Display Forms Facility (Data Dictionary Online Facility).

CHAPTER 3. ADFIN PROCESSOR

The ADFIN processor consists of two phases:

- ADFX Rules Generator Extract Processing
- ADFIN Data Dictionary Format Processing

ADFX RULES GENERATOR EXTRACT PROCESSOR

DESCRIPTION

The first step in the ADFIN process is to extract data from the Rules Generator SYSTEM, SEGMENT, FIELD, and GENERATE statements. This is done by the Rules Generator Extract Processor, which is invoked by the Rules Generator. It creates two output files containing the extracted Rules Generator data. One file is used by the IMSADF II DB/DC Data Dictionary Extension ADFIN Processing. The other file is used to create the ISPF Tables for IADF.

The ADFX processor is invoked by the IMSADF II Rules Generator when OPTIONS=ADFX is coded on the GENERATE statement. Please refer to the _IMS Application Development Facility II Version 2 Release 2 Application Development Reference_ for further information regarding the coding for and execution of the Rules Generator. It should be noted that the data extracted has been edited by the Rules Generator for validity. If conflicting operands have been specified in the Rules Generator input, the appropriate overrides and defaults will be applied by the Rules Generator and will be reflected in the extracted data.

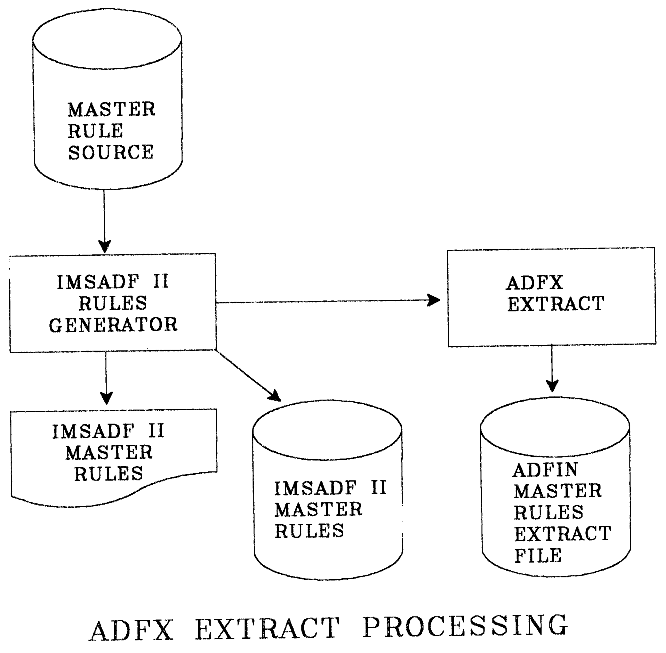

Figure 3-1. IMSADF II ADFX Extract Processing

OPERANDS

The ADFX GENERATE statement has the following operands:

OPTIONS=ADFX	request for ADFX Extract phase processing
DDADFX=	ddname of data set for Extract output for the Data Dictionary; DSORG=PS, LRECL=256 **Note:** The Extract output for the ISPF Tables ALWAYS has a ddname of ISPF, DSORG=PS, LRECL=1200
SEG=(list)	IMSADF II SEGIDs for those segments to be processed by ADFX for the Data Dictionary. If not specified, all data base segments will be extracted for the Data Dictionary. **Note:** All segments (data base or otherwise) are ALWAYS extracted for the ISPF tables
DECKS=<u>NO</u> YES	Sets the DECKS operand for the remainder of the Rules Generator run. When DECKS=YES is specified, the generated Assembler Language rules source will be output to a data set with ddname=GENDECK. DECKS=NO indicates that object decks are to be created and link-edited in the appropriate rules load library. The default is NO.

Note: The placement of the OPTIONS=ADFX GENERATE statement within the Rules Generator source is VERY important.

1. The ADFX Processor will only extract SYSTEM, SEGMENT, and FIELD statement information from those statements which **precede** the OPTIONS=ADFX GENERATE statement statement in the Rules Generator source.

2. The ADFX Processor will only extract GENERATE statement information from those statements which are **after** the OPTIONS=ADFX GENERATE statement in the Rules Generator source.

EXAMPLES

These are several examples of Rules Generator source which invoke the ADFX Extract Processor. Following each example is a description of the resulting Extract processing.

Example 1

```
SYSTEM   SYSID=MFC1,SOMTX=XX,PGROUP=XX,DECKS=NO
SEGMENT  ID=AA,TYPE=PS
FIELD    ID=AA01,LEN=2
FIELD    ID=AA02,LEN=3
SEGMENT  ID=1A,TYPE=DBS,NAME=SEG1A,PARENT=0,LEN=14,KEYNAME=KEY1A
FIELD    ID=1A01,LEN=2,KEY=YES
FIELD    ID=1A02,LEN=6
SEGMENT  ID=2A,TYPE=DBS,NAME=SEG2A,PARENT=1A,LEN=10,KEYNAME=KEY2A
FIELD    ID=2A01,LEN=4,KEY=YES
FIELD    ID=2A02,LEN=2
SEGMENT  ID=3A,TYPE=DBS,NAME=SEG3A,PARENT=2A,LEN=12,KEYNAME=KEY3A
FIELD    ID=3A01,LEN=6,KEY=YES
FIELD    ID=3A02,LEN=3
SEGMENT  ID=M4,TYPE=OUT
FIELD    KWNAME=SPADATE
GENERATE OPTIONS=ADFX,DDADFX=ADFXDD,SEG=(1A,3A),DECKS=YES
GENERATE SEGMENTS=AA,OPTIONS=SEGL
GENERATE SEGMENTS=(1A,2A,3A),OPTIONS=SGALL
GENERATE TRXID=2A,DBPATH=2A,TRXNAME='ADFX TRANS',
         OPTIONS=(CVALL),TSEGS=(AA,1A,2A)
/*
```

In this ADFX Extract run, the Extract file for the Data Dictionary would contain information from the SYSTEM statement, the SEGMENT statements for segment IDs 1A and 3A, and the FIELD statements for field IDs 1A01, 1A02, 3A01, and 3A02. The Extract file for the ISPF Tables would contain information from the SYSTEM statement, the SEGMENT statements for segment IDs AA, 1A, 2A, 3A, M4, the FIELD statement information for field IDs AA01, AA02, 1A01, 1A02, 2A01, 2A02, 3A01, 3A02, and the KWNAME field, and the GENERATE statement information from the three GENERATE statements which follow the OPTIONS=ADFX GENERATE statement. The DECKS=YES on the OPTIONS=ADFX GENERATE statement would override the DECKS=NO on the SYSTEM statement.

Example 2

```
SYSTEM   SYSID=MFC1,SOMTX=XX,PGROUP=XX
SEGMENT  ID=AA,TYPE=PS
FIELD    ID=AA01,LEN=2
FIELD    ID=AA02,LEN=3
SEGMENT  ID=1A,TYPE=DBS,NAME=SEG1A,PARENT=0,LEN=14,KEYNAME=KEY1A
FIELD    ID=1A01,LEN=2,KEY=YES
FIELD    ID=1A02,LEN=6
SEGMENT  ID=2A,TYPE=DBS,NAME=SEG2A,PARENT=1A,LEN=10,KEYNAME=KEY2A
FIELD    ID=2A01,LEN=4,KEY=YES
FIELD    ID=2A02,LEN=2
SEGMENT  ID=3A,TYPE=DBS,NAME=SEG3A,PARENT=2A,LEN=12,KEYNAME=KEY3A
FIELD    ID=3A01,LEN=6,KEY=YES
FIELD    ID=3A02,LEN=3
SEGMENT  ID=M4,TYPE=OUT
FIELD    KWNAME=SPADATE
GENERATE OPTIONS=ADFX,DDADFX=ADFXDD,DECKS=YES
GENERATE SEGMENTS=AA,OPTIONS=SEGL
GENERATE SEGMENTS=(1A,2A,3A),OPTIONS=SGALL
GENERATE TRXID=2A,DBPATH=2A,TRXNAME='ADFX TRANS',
         OPTIONS=(CVALL),TSEGS=(AA,1A,2A)
/*
```

In this ADFX Extract run, the Extract file for the Data Dictionary would contain information from the SYSTEM statement, the SEGMENT statements for segment IDs 1A, 2A and 3A, and the FIELD statements for field IDs 1A01, 1A02, 2A01, 2A02, 3A01 and 3A02. The Extract file for the ISPF Tables would contain information from the SYSTEM statement, the SEGMENT statements for segment IDs AA, 1A, 2A, 3A, M4, the FIELD statement information for field IDs AA01, AA02, 1A01, 1A02, 2A01, 2A02, 3A01, 3A02, and the KWNAME field, and the GENERATE statement information from the three GENERATE statements which follow the OPTIONS=ADFX GENERATE statement. The DECKS=YES on the OPTIONS=ADFX GENERATE statement would set the DECKS operand for the remainder of the Rules Generator run.

Example 3

```
SYSTEM  SYSID=MFC1,SOMTX=XX,PGROUP=XX
SEGMENT ID=AA,TYPE=PS
FIELD   ID=AA01,LEN=2
FIELD   ID=AA02,LEN=3
SEGMENT ID=1A,TYPE=DBS,NAME=SEG1A,PARENT=0,LEN=14,KEYNAME=KEY1A
FIELD   ID=1A01,LEN=2,KEY=YES
FIELD   ID=1A02,LEN=6
GENERATE SEGMENTS=AA,OPTIONS=SEGL
GENERATE OPTIONS=ADFX,DDADFX=ADFXDD,DECKS=NO
SEGMENT ID=2A,TYPE=DBS,NAME=SEG2A,PARENT=1A,LEN=14,KEYNAME=KEY2A
FIELD   ID=2A01,LEN=4,KEY=YES
FIELD   ID=2A02,LEN=2
SEGMENT ID=M4,TYPE=OUT
FIELD   KWNAME=SPADATE
GENERATE SEGMENTS=1A,OPTIONS=SGALL
GENERATE TRXID=1A,DBPATH=1A,TRXNAME='ADFX TRANS',
         OPTIONS=(CVALL),TSEGS=(AA,1A,2A)
/*
```

In this ADFX Extract run, the Extract file for the Data Dictionary would contain information from the SYSTEM statement, the SEGMENT statement for segment ID 1A, and the FIELD statements for field IDs 1A01 and 1A02. The Extract file for the ISPF Tables would contain information from the SYSTEM statement, the SEGMENT statements for segment IDs AA and 1A, the FIELD statement information for field IDs AA01, AA02, 1A01 and 1A02, and the GENERATE statement information from the two GENERATE statements which follow the OPTIONS=ADFX GENERATE statement. The DECKS=NO on the OPTIONS=ADFX GENERATE statement would set the DECKS operand for the remainder of the Rules Generator run. No information would be extracted for segment IDs 2A or M4, or any of their fields, since these SEGMENT statements are after the OPTIONS=ADFX GENERATE statement. Also, no information would be extracted for the GENERATE statement with SEGMENTS=AA,OPTIONS=SEGL since this GENERATE statement occurred before the OPTIONS=ADFX GENERATE statement.

ADFIN DATA DICTIONARY FORMAT PROCESSOR

The ADFIN Format Processor uses the data extracted from the Rules Generator source to create Data Dictionary batch commands. These batch commands will update the IMSADF II extensibility subjects using the information from the Rules Generator source.

DESCRIPTION

The ADFIN Format processor uses the Data Dictionary's Program Access Facility (PAF) to verify the data contained in the Data Dictionary data bases that make up the model.

The Program Access Facility allows users to write their own programs to access the data in the Data Dictionary data bases in order to produce customized output. The program retrieves Dictionary data by calling the Dictionary with its request. The program takes advantage of the data base retrieval mechanisms already present in the Dictionary system. The program is not dependent on internal Dictionary logic or data base structures. The Dictionary retrieves the requested data and places it in a data area belonging to the program.

The ADFIN Format processor must be link-edited with the Data Dictionary-supplied assembler interface module (DBDWLNKA) and must reside in a load library accessible to the Data Dictionary. The ADFIN Format processor link-edit is supplied with the base IMSADF II program product. However, the link-edit does not occur automatically when IMSADF II is installed. Refer to the IMS Application Development Facility II Version 2 Release 2 Installation Guide for information on the two additional steps that are required to install the ADFIN Format processor.

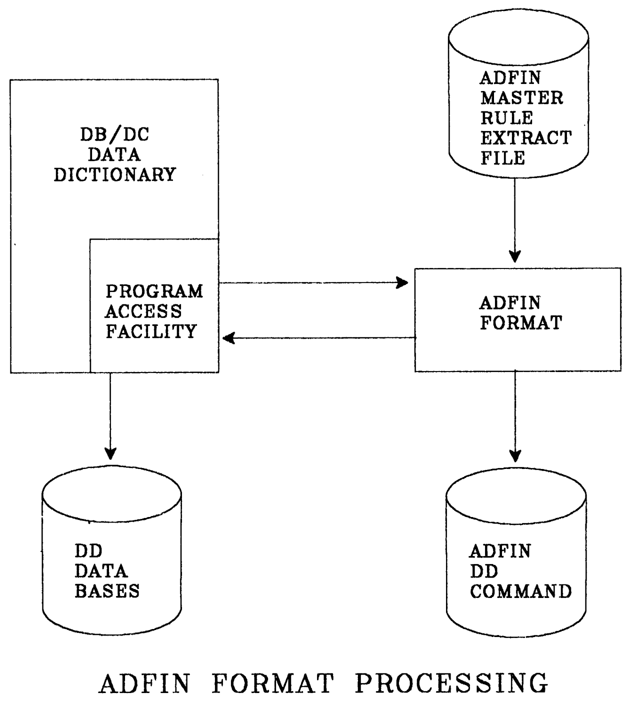

Figure 3-2. IMSADF II ADFIN Format Processing

ADFIN EXECUTE COMMAND

The ADFIN Format processor can be invoked for batch or online use with the Data Dictionary EXECUTE command. The Command Screen, from the Interactive Display Forms Facility, can be used to invoke the EXECUTE command or it can be part of a batch input stream. The EXECUTE command specifies the program name and passes up to 200 characters of data to the program. The command format is:

```
 EXECUTE PGM=????Y09 PARM='PARAMETER STRING';
```

The question marks should be changed to the four-character application system ID under which the Data Dictionary Extension is installed. (See the IMS Application Development Facility II Version 2 Release 2 Installation Guide).

Example:

```
 EXEC PGM=MFC1Y09 +
      PARM='UPDATE=YYY,PSBNAME=STDPSBNM,DDADFX=DADFX,DDADFCMD=DADFCMD';
```

A summary of the parameters on the EXECUTE command follows.

PARAMETERS

A parameter string of up to 200 bytes of character data is defined on the EXECUTE command. If a single quotation mark is to appear in the string, it must be specified as two single quotation marks. Standard Data Dictionary command continuation rules apply when specifying a long parameter string.

- To continue a D/D parameter
  - The parameter must begin with a single quote and end with a single quote

- Put + at the end of the line to be continued (do not leave a space between the comma and the plus sign)

- Start the following line in column 1; otherwise, all columns from 1 to the column where you do start will be passed in the parameter string as blanks

• To continue a D/D command

  - The continued line does not have to start in column 1

  - The + that signals continuation must be preceded by a space. If the + is not preceded by a space, the D/D will issue a misleading error message that the parameter is too long

Example:
```
    EXEC PGM=MFC1Y09  +
    PARM='UPDATE=YYY,PSBNAME=STDPSBNM,+
    DDADFX=DADFX,DDADFCMD=DADFCMD';
```

All parameters are treated as keywords. Each keyword must be followed by an equal sign and its value. A comma is the only valid delimiter between keywords. Blanks are not valid, except after the last parameter value. The parameters are as follows:

The ADFIN Format program processing is controlled by the following parameters:

UPDATE=YYY YNY YYN YNN NNN NNY NYY NYN	Specifies whether or not to update the subjects in ADFSYS01, ADFSEG01, and ADFDTE01 categories, respectively, with Y=yes and N=no. If the subject is found, the default is N. If the subject is not found, the default value is Y.
STDSTAT=	One-character Dictionary status code of subjects in standard categories of the IMSADF II model; default value is P. Refer to the <u>Data Dictionary Terminal User's Guide</u> for the status code values.
ADFSTAT=	One-character Dictionary status code of subjects in extensibility categories of the IMSADF II model; default value is P. Refer to the <u>Data Dictionary Terminal User's Guide</u> for the status code values.
STDOCC=	Three-character Dictionary occurrence number of subjects in standard categories of the IMSADF II model; default value is 0. Refer to the <u>Data Dictionary Terminal User's Guide</u> for the occurrence number values.
ADFOCC=	Three-character Dictionary occurrence number of subjects in extensibility categories of the IMSADF II model; default value is 0. Refer to the <u>Data Dictionary Terminal User's Guide</u> for the occurrence number values.
TRAILER=	One-character transaction trailer
PGMID=	Two-character program id; matches cluster code; default value is value of SOMTX keyword on SYSTEM statement of Rules Generator Source.

PSBTYPE=	One-character PSB type:
	C = conversational
	N = nonconversational
	S = special processing routine-unclustered
	B = batch
	default value is C.

PSBNAME=	Eight-character PSBNAME
DDADFX=	The ddname of data set for Extract output; DSORG=PS, LRECL=256; default value is DDADFX.
DDADFCMD=	The ddname of data set for Format output; Dictionary update commands; DSORG=PS, LRECL=80; default value is DDADFCMD.
SIGNON=	Sign-on id for installations with security; one to 31 characters in length.
PASSWORD=	Eight-character password for installations with security; if specified, must not be all blanks.
FLUSH=<u>YES</u> NO	Allows the user to specify if he wishes the dictionary to stop processing if an error occurs; default is YES.

- The UPDATE parameter also controls the updating of the extensibility relationships. An extensibility relationship will be updated only if any extensibility subjects involved in the relationship are to be updated. For example, UPDATE=NYY will cause the ADFSEG01 and ADFDTE01 subjects to be updated, but not the ADFSYS01 subject. It also cause the POINT_SEG, POINT_DTE, and ADFSEG_HAS relationships to be updated. The POINT_DBS and ADFSYS_HAS relationships will not be updated since they involve the ADFSYS01 subject.

- TRAILER, PGMID, PSBTYPE, and PSBNAME are used to determine the PSB subject name. If none of these are specified, PSB subject name is based on values found in ADFOPTN and the SYSID on the SYSTEM statement. The PSB subject name is in the ssssTxxa Format, as explained in the <u>IMS Application Development Facility II Version 2 Release 2 Application Development Reference</u>.

- If PSBNAME is specified, it is used in processing and the values of TRAILER, PGMID, and PSBTYPE will be ignored.

- If PSBNAME is not specified and any of the other optional parameters are, they are used for determining a non-default PSBNAME.

LOGIC FLOW

The ADFIN Format processor program and the DB/DC Data Dictionary communicate using a control area, a format table and a data area. These control blocks are in the ADFIN Format processor program and can be accessed and updated by both the ADFIN Format processor program and the Data Dictionary. Retrieval and output requests from the ADFIN Format processor to the Data Dictionary are initiated by CALL statements.

When the ADFIN Format processor is invoked by the Data Dictionary, it is passed the parameter string from the EXECUTE command with no modification, except for removing the leading and trailing single quotation marks and for reducing pairs of single quotation marks to a single quotation mark.

The ADFIN Format processor analyzes the input parameters. If no errors are found in the input parameters, the data from the ADFX Rules Generator Extract file is used to create DB/DC Data Dictionary batch commands. Program Access Facility calls are made to verify the subjects in the Standard PSB, PCB, DATABASE, SEGMENT, and ELEMENT categories which correspond to the subjects in the IMSADF II extensibility categories. The Data Dictionary commands are written to the data set whose DDNAME was specified by the DDADFCMD parameter of the EXECUTE command.

When the ADFIN Format processor completes and returns control to the Data Dictionary, it sets a return code for the Data Dictionary to examine. Each execution of the ADFIN Format processor generates at least one message.

Return Code	Meaning
0	Processing successfully completed, informational message issued.
4	Execution continued to completion, one or more warning messages issued.
8	Processing terminated prior to completion, error message issued, error should be correctable by the user.
12	Severe error, only occurs when routing output through the Data Dictionary fails. Program terminates, no error message is issued.

ADFIN Format processor messages are documented in Chapter 5, "Messages."

OUTPUT LOGIC

The ADFIN Format processor creates Data Dictionary ADD, CHANGE_IN, and ADD_RELATIONSHIP commands for the subjects in the IMSADF II extensibility categories which are allowed to be updated as specified by the UPDATE parameter value. These commands are written to the data set referenced by the ddname specified in the DDADFCMD parameter. If the DDADFCMD parameter is not specified in the parm list, a default ddname of DDADFCMD will be used. The ADFIN output file is to be used as input to the DB/DC Data Dictionary Batch Procedure. If the IMSADF II extensibility subjects already exist in the Data Dictionary, the DB/DC Data Dictionary Batch Procedure will issue a return code of 4. This is due to the attempt to add one or more subjects and relationships which already exist in the Dictionary. This does not mean that there has been an error in the ADFIN or Data Dictionary processing; it should be regarded as an informational message. The ADFIN Format output data set, containing Data Dictionary batch commands, should be used as the SYSIN data set in the Data Dictionary standard batch procedure.

Note: During the time between running ADFIN Format and running the Data Dictionary standard batch procedure, NO modifications should be made to the Data Dictionary.

TRXNAME

The ADFX Extract Processor does not extract the TRXNAME from the IMSADF II Rules Generator GENERATE statement. The ADFIN Format Processor does not add the TRXNAME to the first descriptor segment of the ADFSEG01 subject. The user can update the first descriptor segment of the ADFSEG01 subject by using a Data Dictionary CHANGE_IN command or the Data Dictionary Interactive Display Forms Facility. This is only necessary if the user wishes the TRXNAME keyword to be included in the default transactions generated by ADFOUT.

ADFIN IMSADF II FIELD PROCESSING

While processing each IMSADF II FIELD from the Rules Generator source, the ADFIN processor attempts to locate a standard Data Dictionary ELEMENT subject which corresponds to the IMSADF II FIELD. This is necessary for the creation of an ADD_RELATIONSHIP command for the POINT_DTE relationship. The following requirements must be met by the standard ELEMENT subject:

- It must be related to the standard SEGMENT subject which corresponds to the IMSADF II SEGMENT which contains the current IMSADF II FIELD.

- It must have a status code equal to the value of the STDSTAT parameter on the EXEC statement. If STDSTAT was not specified, its default value is P.

- It must have an occurrence number equal to the value of the STDOCC parameter on the EXEC statement. If STDOCC was not specified, its default value is 0.

- It must have an appropriate TYPE attribute value which corresponds to the TYPE keyword value on the IMSADF II Rules Generator FIELD statement. The following table indicates which TYPE values correspond and when it is necessary to override the standard ELEMENT TYPE attribute with a value in the OTYPE attribute of the ADFDTE01 subject.

IMSADF II FIELD Statement TYPE Keyword Value	Data Dictionary Standard ELEMENT TYPE Attribute	Data Dictionary ADFDTE01 OTYPE Attribute
Alpha	C	Alpha
Alphanum	C	
Num	C or Z *	Num
Date	C or Z **	Date
Dec	Z	
Pd	P	
Bin	H or F	
Bit	B	
Hex	X or E	
Dbcs	C	Dbcs
Mixed	C	Mixed
Varchar	C ***	
Float	D	

* No decimal point in COBPIC or PLIPIC attribute values.
** No decimal point in COBPIC or PLIPIC attribute values and BYTES attribute must have a value of 6.
*** Scale = varying

Figure 3-3. ADFDTE01 OTYPE Values

- The value of the BYTES attribute of standard ELEMENT must be the same as the value of the BYTES or LENGTH keyword of the Rules Generator FIELD statement.

- The START attribute of the WITH relationship between the standard SEGMENT and the standard ELEMENT must correspond to the value of the START or POSITION keyword of the Rules Generator FIELD statement. In the case that the standard ELEMENT is an array, additional checking is done to see if the IMSADF II FIELD corresponds to any of the members of the array. A standard ELEMENT subject is considered to be an array if the OCCURS or any of the DIM attributes are non-blank. If the OCCURS attribute is used, the first non-blank character specified must be the numeric character(s) specifying the array size. Only the first five numeric characters are processed. If the PLI dimension attributes are used, both upper and lower bounds on up to three dimensions are processed to determine array size. ADFIN supports up to three levels of ELEMENTs which are arrays. For example:

```
01 Array-record.
   03 Array-item-1 occurs 10 times.
      05 Array-item-2 occurs 4 times.
         07 Array-item-3 occurs 7 times.
```

  ADFIN only supports subelements in single-dimensional arrays. For example:

```
01 Array-record.
   03 Array-group-item occurs 10 times.
      05 Subelement-1 pic X(20).
      05 Subelement-2 pic XX.
```

ADFIN does not support subelements in multi-dimensional arrays. For example, the following is NOT supported:

```
01 Array-record.
   03 Array-group-item-1 occurs 10 times.
      05 Array-group-item-2 occurs 4 times.
         07 Subelement-1 pic X(20).
         07 Subelement-2 pic XX.
```

If all the above requirements have been met, an ADD_RELATIONSHIP command is created by ADFIN for the POINT_DTE relationship between the ADFDTE01 subject and the standard ELEMENT subject. If no standard ELEMENT subject can be found which meets all of the requirements, a warning message is printed, and no updates are made for the current segment. Processing will continue for the next segment.

CHAPTER 4. ADFOUT PROCESSOR

The ADFOUT processor uses the Data Dictionary's Program Access Facility (PAF) to access the data contained in the Data Dictionary data bases that make up the model.

The Program Access Facility allows users to write their own programs to access the data in the Data Dictionary data bases in order to produce customized output. The program retrieves Dictionary data by calling the Dictionary with its request. The program takes advantage of the data base retrieval mechanisms already present in the Dictionary system. The program is not dependent on internal Dictionary logic or data base structures. The Dictionary retrieves the requested data and places it in a data area belonging to the program.

The ADFOUT processor must be link-edited with the Data Dictionary-supplied assembler interface module (DBDWLNKA) and must reside in a load library accessible to the Data Dictionary. The ADFOUT processor link-edit is supplied with the base IMSADF II program product. However, the link-edit does not occur automatically when IMSADF II is installed. Refer to the IMS Application Development Facility II Version 2 Release 2 Installation Guide, for information on the two additional steps that are required to install the ADFOUT processor.

The ADFOUT processor can be invoked for batch or online use with the Data Dictionary EXECUTE command. The Command Screen of the Interactive Display Forms Facility can be used to invoke the EXECUTE command or it can be part of a batch input stream. The EXECUTE command specifies the program name and passes up to 200 characters of data to the program. The command format is:

```
                 EXECUTE PGM=????Y01 PARM='PARAMETER STRING';
```

The question marks should be changed to the four-character application system ID within which the Data Dictionary Extension is defined (See the IMS Application Development Facility II Version 2 Release 2 Installation Guide).

Example:

```
        EXEC PGM=MFC1Y01 +
             PARM='SYS=SAMP,STA=T,SEG=ALL,RUL=YES,TRX=YES,DES=L';
```

A summary of the parameters on the EXECUTE command follows.

PARAMETERS

A parameter string of up to 200 bytes of character data is defined on the EXECUTE command. If a single quotation mark is to appear in the string, it must be specified as two single quotation marks. Standard Data Dictionary command continuation rules apply when specifying a long parameter string.

- To continue a D/D parameter
  - The parameter must begin with a single quote and end with a single quote
  - Put + at the end of the line to be continued (do not leave a space between the comma and the plus sign)
  - Start the following line in column 1; otherwise, all columns from 1 to the column where you do start will be passed in the parameter string as blanks.
- To continue a D/D command
  - The continued line does not have to start in column 1

- The + that signals continuation must be preceded by a space. If the + is not preceded by a space, the D/D will issue a misleading error message that the parameter is too long.

  Example:

```
     EXEC PGM=MFC1Y01  +
     PARM='SYS=SKIL,STA=,+
     INC=YES,RUL=YES,TRX=YES,SEG=(SK)';
```

All parameters are treated as keywords. Each keyword must be followed by an equal sign and its value. A comma is the only valid delimiter between keywords. Blanks are not valid, except after the last parameter value. Allowable abbreviations are the first three characters of the keyword. The parameters are as follows:

SYSID=
SYS=
: The IMSADF II application system ID. The user name portion of the subject-name in the ADFSYS01 category. This parameter is required.

STATUS=
STA=
: The Data Dictionary status code for the system ID. The status code indicates whether the system ID is in production status or in test status (A-T,0-9). The default is 'P'.

OCC=
: The Data Dictionary occurrence number for the system ID. The occurrence number can be used to differentiate between otherwise identical system IDs. Valid range is 000-255. The default is '000'.

INCLUDE=
INC=
: YES or NO. Specifies whether to create source INCLUDE members. The default is YES.

RULE=
RUL=
: YES or NO. Specifies whether to create segment rule source. The default is 'NO'. If YES, create IMSADF II SYSTEM statement, data base SEGMENT and FIELD statements, and a GENERATE OPTION=SGALL statement.

TRX=
: YES or NO. Specifies whether to create default transaction source. The default is 'NO'. If YES, create IMSADF II SYSTEM statement, data base SEGMENT and FIELD statements, and GENERATE OPTION=CVALL statement(s). If TRX=YES is specified, an additional GENERATE statement is appended at the end (GENERATE OPTION=SOMSS) to create or update the conversational Secondary Option Menu rule. The transaction ID used for all default transactions is the two-character IMSADF II segment ID.

  The INCLUDE, RULE or TRX parameter must be specified as YES (INCLUDE defaults to YES). If none of these parameters is YES, the type of output has not been defined and the ADFOUT processor terminates with an error message. TRX=YES and RUL=YES create the same output except for the GENERATE statements.

SEG=
: ALL or a list of IMSADF II segment IDs in parentheses delimited with a comma. The default is 'ALL'. This parameter defines what IMSADF II data base segment and field definitions are to be included in the output source. If ALL, the ADFOUT processor creates data base SEGMENT and FIELD statements for every IMSADF II segment related to the specified application system ID. If the list option is used a maximum of ten IMSADF II segment IDs can be specified. Parent IMSADF II segment IDs need not be specified. If the parent definition is required (when RULE=YES or TRX=YES are specified for non-root segments), the IMSADF II parent segment and field definitions are included for each level back to the root level.

PGROUP=
PGR=
: Specifies a two-character IMSADF II project group ID. This parameter is used to override the PGROUP attribute in the ADFSYS01 category which is the default.

SOMTX=
: Specifies a two-character IMS/VS transaction cluster code.

SOM=	This parameter is used to override the SOMTX attribute in the ADFSYS01 category which is the default. The Rules Generator requires the PGROUP and SOMTX parameters when generating input transaction rules. If default transactions are generated with this output, TRX=YES specified, the PGROUP and SOMTX parameters should be defined either as ADFSYS01 attributes or as parameters.
DEST= DES=	Output destination. DEST can be equal to T, L, or P. T is for terminal, L is for printer, and P is for punch. If DEST= is not specified, the output is routed to the data sets referenced by the DINCLUDE and DADFOUT DD cards, unless the DINCLUDE= and DADFOUT= parameters are specified. Refer to these parameters and to "Output Routing" on page 4-4 for further information.
IEB=	YES or NO. Specifies whether to create IEBUPDTE function statements. The default is YES. This parameter is ignored unless INC=YES is also specified or assumed by default. If IEB=YES, the IEBUPDTE Function statements are inserted into the output file created when INC=YES is specified.
DINCLUDE= DIN=	The ddname of user-specified data set for INC=YES output. Data set has LRECL=80 and is physical sequential or partitioned. For a partitioned data set, the member name is specified in the DSNAME on the added ddcard or is specified using the DINMBR= parameter. DINCLUDE= cannot be specified when DEST= is specified.
DINMBR=	Member of partitioned data set referenced by the DINCLUDE ddcard or by the ddcard with the ddname specified in the DINCLUDE= parameter.
DADFOUT=	The ddname of user-specified data set for RUL=YES and TRX=YES output. Data set has LRECL=80 and is physical sequential or partitioned. For a partitioned data set, the member is specified in DSNAME on the added ddcard or is specified using the DADMBR= parameter. DADFOUT cannot be specified when DEST= is specified.
DADMBR=	Member of the partitioned data set referenced by the DADFOUT DDCARD or by the DDCARD with the DDNAME specified in the DADFOUT=parameter.

LOGIC FLOW

The ADFOUT processor program and the DB/DC Data Dictionary communicate using a control area, a format table and a data area. All three of these control blocks are in the ADFOUT processor program and can be accessed and updated by both the ADFOUT processor program and the Data Dictionary. Retrieval and output requests from the ADFOUT processor to the Data Dictionary are initiated by CALL statements.

When the ADFOUT processor is invoked by the Data Dictionary, it is passed the parameter string from the EXECUTE command with no modification, except for removing the leading and trailing single quotation marks and for reducing pairs of single quotation marks to a single quotation mark.

The ADFOUT processor analyzes the input parameters. If no errors are found in the input parameters, data is retrieved from the Data Dictionary based on the input. The retrieved data is processed into Rules Generator source statements. These statements are routed to the appropriate output destination.

When the ADFOUT processor completes and returns control to the Data Dictionary, it sets a return code for the Data Dictionary to examine. Each execution of the ADFOUT processor generates at least one message.

Return Code	Meaning
0	Processing successfully completed, informational message issued.
4	Execution continued to completion, one or more warning messages issued.
8	Processing terminated prior to completion, error message issued, error should be correctable by the user.
12	Severe error, only occurs when routing output through the Data Dictionary fails. Program terminates, no error message is issued.

All ADFOUT processor messages are documented in Chapter 5, "Messages."

OUTPUT LOGIC

Output control information is passed to the ADFOUT processor via the parameter string on the EXECUTE command. Output type is controlled by INCLUDE, IEB, RULE, and TRX. Output destination is controlled by DEST, DINCLUDE, DADFOUT, and the Dictionary SETPRINT and SETPUNCH commands, which may be issued prior to the execution of ADFOUT.

OUTPUT ROUTING

Output can be routed to the online user who invoked the ADFOUT processor or to data sets. These data sets are as follows:

DDLIST	ddname of printer allocated to the Data Dictionary.
DDPUNCH	ddname of punch allocated to the Data Dictionary.
DINCLUDE	ddname of physical sequential or partitioned data set with LRECL=80 used for INC=YES output; default ddname is DINCLUDE.
DADFOUT	ddname of physical sequential or partitioned data set with LRECL=80 used for TRX=YES and RUL=YES output; default ddname is DADFOUT.
ddname1	user-specified ddname of physical sequential or partitioned sequential data set with LRECL=80 used for INC=YES output; any valid ddname may be specified.
ddname2	user-specified ddname of physical sequential or partitioned sequential data set with LRECL=80 used for TRX=YES and RUL=YES output; any valid ddname may be specified.

The data sets referenced by DINCLUDE, DADFOUT, ddname1, and ddname2 are under the control of the ADFOUT processor. If used, these DD cards must be added to the Data Dictionary batch processing JCL or, if the ADFOUT processor is executing online, to the IMS/VS message region JCL. Before issuing an open to these data sets, the ADFOUT processor determines if a valid data set name was specified. If the data set names are specified as DD DUMMY or if the DD cards are not in the JCL, an error message is issued and ADFOUT terminates with a return code of 8. If valid data sets are specified, the OPEN is issued and the data sets are used for output. Before the ADFOUT processor terminates, the DINCLUDE, DADFOUT, ddname1, and ddname2 data sets are closed, if previously opened.

DDLIST and DDPUNCH output are produced by the ADFOUT processor through a Program Access Facility call to the Data Dictionary. The ADFOUT processor has no control of the output routing once the call has been issued. If the call fails, the ADFOUT processor terminates with a return code of 12.

The DEST parameter determines the actual destination of ADFOUT output. DEST functions in the following manner:

- When DEST is specified

  DEST=T All output is routed to the online terminal that invoked ADFOUT. If Data Dictionary is running as a BMP or in batch, output is routed to DDLIST.

  DEST=L All output is routed to DDLIST.

  DEST=P All output is routed to DDPUNCH.

  If the DEST parameter is specified as T, P, or L and INC=YES as well as RUL=YES or TRX=YES, the INCLUDE output is superseded.

- When DEST is not specified

  INC=YES output is routed to the data set referenced by DINCLUDE unless DINCLUDE=ddname1 is specified. The ddname1 is any valid ddname. If DINCLUDE=ddname1 is specified, the INC=YES output is routed to the data set referenced by ddname1.

  RUL=YES and TRX=YES output are routed to the data set referenced by DADFOUT, unless DADFOUT=ddname2 is specified. The ddname2 is any valid ddname. If DADFOUT=ddname2 is specified, the RUL=YES and TRX=YES output are routed to the data set referenced by ddname2.

  If DINCLUDE, DADFOUT, ddname1, and ddname2 are partitioned sequential data sets, members to be updated are referenced by one of two methods:

  The member name is included as part of the data set name.

  DINMBR=member1 and DADMBR=member2 are specified in the parameter list. DINMBR is used with DINCLUDE or ddname1 and DADMBR is used with DADFOUT or ddname2.

SETPRINT and SETPUNCH utilize a JOB parameter to reference a Dictionary JOB subject whose userdata contains JCL. If SETPUNCH routes output to the internal reader, a job is submitted using the userdata JCL with the ADFOUT output as the SYSIN data. This method is used to submit a batch Rules Generator or IEBUPDTE job from the Dictionary run.

To submit a batch Rules Generator job from the Data Dictionary:

1. In SYSUSERn of a subject JOB (s,J,username,0), enter

   ```
   valid job card
   Rules Generator JCL
   //*ROUTED OUTPUT(indicates where Rules Generator source
                              should be inserted)
   ```

2. Add the following ddcard, or equivalent, to the Data Dictionary JCL or the the message region

   ```
   //DDINTRDR DD SYSOUT=(A,INTRDR)
   ```

   to enable output to be sent to the MVS internal reader. The ddname may be any valid ddname that begins with the two characters 'DD'.

3. Issue the command

   ```
   SETPUNCH DDOUT=DDINTRDR          JOB=(s,J,username,0) UDNO=n;
   ```

   where DDINTRDR is the ddname of the ddcard added in step 2, (s,J,username,0) is the JOB subject whose userdata contains the Rules Generator JCL, and n is the number of userdata that contains this JCL.

4. Execute the ADFOUT processor with DEST=p.

Assuming the JOB subject referenced in Step 1 is named (T,J,RULGEN,0), and the ddcard referenced in Step 2 is DDINTRDR, message output from the Data Dictionary and the ADFOUT processor is as follows, based on the sample problem distributed with the IMSADF II.

```
15:53:28 04/01/84 INPUT RECORD  /SETPUNCH DDOUT=DDINTRDR JOB=(T,J,RULGEN,0);

DBD0086 I DDOUT=DDINTRDR
DBD0086 I JOB=TJ RULGEN 0 UDNO=1
15:53:28 04/01/84 INPUT RECORD /PARM='SYS=SAMP,STA=T,SEG=ALL,RULE=YES,TRX=YES,DES=P';
ADF1031 I PROCESSING SUCCESSFULLY COMPLETED
DBD0094 I OUTPUT COMPLETE FOR DDNAME DDINTRDR
```

To submit a batch IEBUPDTE job from the Data Dictionary:

Submitting a batch IEBUPDTE job from the Data Dictionary is similar to submitting a Batch Rules Generator job. The same steps are followed, with IEBUPDTE JCL added to the JOB subject userdata.

Sample commands to add IEBUPDTE JCL, route the output, and execute the ADFOUT processor follow.

```
ADD JOB (T,J.IEBUPDTE,0) SYSUSER1=(10,'//IEBJOB JOB (ACCT #),MSGLEVEL=1,CLASS=A');
A JOB" SYSUSER1=(20,'//S1  EXEC PGM=IEBUPDTE,PARM=NEW');
A JOB" SYSUSER1=(30,'//SYSUT1 DD  DSN=USER.IMSADF.INCLUDE,DISP=SHR');
A JOB" SYSUSER1=(40,'//SYSUT2 DD  DSN=USER.IMSADF.INCLUDE,DISP=SHR');
A JOB" SYSUSER1=(50,'//SYSPRINT DD SYSOUT=A')';
A JOB" SYSUSER1=(60,'//SYSIN DD  *');
A JOB" SYSUSER1=(660,'//*ROUTED OUTPUT');
SETPUNCH DDOUT=DDINTRDR JOB=(T,J,IEBUPDTE,0);
EXEC PGM=MFC1Y01  PARM='SYS=SAMP,STA=T,SEG=ALL,INC=YES,DES=P';
```

Refer to the Data Dictionary Terminal User's Guide for further discussion of SETPRINT and SETPUNCH.

Note: All messages issued by ADFOUT are routed with a call to the Data Dictionary through the Program Access Facility. All messages are assigned DEST=T routing.

OUTPUT TYPES

INCLUDE

INCLUDE output consists of IEBUPDTE function statements and IMSADF II SEGMENT and FIELD statements for each IMSADF II segment definition processed by ADFOUT. The name parameter on each IEBUPDTE function statement is made up of the four-character IMSADF II system id and the two-character IMSADF II segment id associated with the segment being processed.

INCLUDE output may be routed to the terminal, the printer assigned to the Data Dictionary, the punch assigned to the Data Dictionary, a sequential data set (DSORG=PS or PO), or the internal reader. See "Output Routing" on page 4-4.

INCLUDE output is superseded if the DEST parameter is specified and RULE=YES or TRX=YES. Under these circumstances, no INCLUDE output is produced.

IF IEB=NO is specified, INCLUDE output is created without IEBUPDTE function statements.

Sample INCLUDE output:

```
./ ADD NAME=SAMPPA,LIST=ALL
./ NUMBER NEW1=1000,INCR=1000
***********************************************************************
*
*  SEGMENT: SEGID= PA     DATE:   09/13/84    TIME:   12:10:16
*           DICTIONARY ADF SEG: T SAMPPA                              000
*           DICTIONARY SEGMENT: TCPARTROOT                            000
*
***********************************************************************
   SEGMENT  ID=PA,
            PARENT=0,
            NAME=PARTROOT,
            LENGTH=00050,
            KEYNAME=PARTKEY,
            SKSEGS=18
*
*           DICTIONARY ADF FLD: T KEY                                 000
*           DICTIONARY   FIELD: TCPART-NUMBER                         000
*
   FIELD    ID=KEY,
            TYPE=ALPHANUM,
            LENGTH=00017,
            POSITION=00001,
            KEY=YES,
            SNAME='PART NUMBER'
(the remaining field statements defined for this segment would follow)
```

For further information on the MVS IEBUPDTE utility, refer to MVS Utilities Manual.

RULE AND TRX

Segment rules source and default transaction source are produced containing IMSADF II SYSTEM statement, SEGMENT and FIELD statements, and GENERATE statements. RULE and TRX output may be directed to the terminal, the printer assigned to the Data Dictionary, the punch assigned to Data Dictionary, a sequential file (DSORG=PS or PO), or the internal reader. Details on routing are found in "Output Routing" on page 4-4.

The only difference between RULE and TRX output is the GENERATE statements. RULE output contains a GENERATE statement with OPTION=SGALL. The GENERATE statement will also have the SEGMENTS= operand if the SEG parameter is specified with the list option.

TRX output includes a GENERATE statement for every data base segment when SEG=ALL is specified. If the list option of the SEG parameter is specified, TRX output includes a GENERATE statement for each segment in the list. TRX output GENERATE statements contain OPTION=CVALL, TRXID, DBPATH, and TRXNAME operands, with TRXID and DBPATH equal to the two-character IMSADF II segment id. After TRX GENERATE statements have been produced for segments, ADFOUT produces a GENERATE statement with OPTION=SOMSS, used to update the Conversational Secondary Option Menu rule.

If the ADFOUT output is passed directly to the Rules Generator:

- KEY=YES operand must be specified on at least one field for every IMSADF II data base segment definition.
- default transaction output must contain PGROUP and SOMTX operands. These operands may either be defined as parameters on the invoking Data Dictionary EXECUTE command or be values in associated ADFSYS01 attributes.

Default transaction screens are more usable when:

- SNAME is defined for each field.

  The SNAME attribute in the ADFDTE01 category contains the IMSADF II field screen name.

- TRXNAME is used.

  TRXNAME is derived from the first descriptor segment of the IMSADF II segment being processed. The first 30 characters of the descriptor segment, starting from the first non-blank character, are used as TRXNAME. If more than 30 characters are encountered, any excess is ignored.

  If the IMSADF II segment has no descriptor segment, the TRXNAME operand is not specified.

After creating data base segment rules, default transaction rules and screens, and a secondary option menu, two additional Rules Generator GENERATE statements must be processed before the default level IMSADF II system becomes executable:

1. GENERATE OPTION=CVSYS statement to create the Sign-on Screen and Primary Option Menu Rule
2. GENERATE OPTION=STLE to link-edit an executable conversational driver.

Neither of the two GENERATE statements listed above is created by ADFOUT.

Refer to Appendix B, "ADFOUT Processor Sample Procedure Output" to review the sample procedure's RULE and TRX output.

Note: The ADFOUT processor processes only an IMSADF II data element whose standard data element is directly related to the appropriate standard segment. Consider the following structure:

```
01  SEG1.
    03  AAA.
        05 BBB pic x.
        05 CCC pic x.
        05 DDD Pic 999V99.
```

BBB, CCC, and DDD are included in the ADFOUT processor output only if they are directly related to SEG1. If they are related to AAA, which is related to SEG1, only AAA is eligible for processing by the ADFOUT processor. An additional requirement for processing an IMSADF II data element related to a standard ELEMENT that is contained within another ELEMENT is:

- the containing standard ELEMENT must have an associated ADFDTE01.

COBOL_IN and PLI_IN automatically relate all standard data elements to the appropriate standard segment. If the standard segment and data element definitions are defined by a different method, then the Dictionary EXTEND_RELATIONSHIP command should be used to relate sub-elements to a standard segment.

ADDITIONAL OPERANDS

The ADFOUT Processor derives additional IMSADF II operands from the standard categories and relationships that are part of the Data Dictionary Extension model. These operands, except for TRXNAME, are derived from Data Dictionary data that should be part of each installations data base definition process. The derived IMSADF II operands are shown in Figure 4-1.

STATEMENT			
OPERANDS	SEGMENT	FIELD	GENERATE
LENGTH	X	X	
PARENT	X		
NAME	X		
KEYNAME	X		
TYPE		X	
POSITION		X	
DECIMAL		X	
BITOFF		X	
TRXNAME			X

Figure 4-1. Additional IMSADF II Operands

The IMSADF II segment LENGTH operand is derived from the Data Dictionary SEGMENT MAXBYTES attribute.

The IMSADF II segment PARENT operand is derived from the Data Dictionary DBS/WITH/SEGMENT PHYPAR relationship attribute.

If PHYPAR attribute is used to retrieve the standard physical parent segment, the first valid IMSADF II segment ID related to the standard physical parent is used to create the IMSADF II Rules Generator PARENT=SEGID operand.

The IMSADF II segment NAME operand is derived from the Data Dictionary SEGMENT subject-name. If the subject-code of the segment is A, the segment subject-name is used as the NAME operand. If the subject-code of the segment is not A, the segments aliases are searched until one is found with a subject-code of A. This alias subject-name is used as the NAME operand. If none of the aliases have a subject-code of A, the first eight characters of the primary segment subject-name are used as the NAME operand, and the ADFOUT processor issues a warning message.

The IMSADF II segment KEYNAME operand is derived from the Data Dictionary SEGMENT/WITH/ELEMENT relationship. All relationships are processed until a field is found that has either a U or an M in the GENFLD relationship attribute. If the subject-code of this field is A, the fields subject-name is used as the KEYNAME operand. If the subject-code of the field is not A, the fields aliases are searched until one is found with a subject-code of A. This alias subject-name is used as the KEYNAME operand. If none of the aliases have a subject-code of A, the first eight characters of the primary field subject-name are used as the KEYNAME operand and the ADFOUT processor issues a warning message.

The IMSADF II field LENGTH operand is derived from the Data Dictionary ELEMENT BYTES attribute.

The IMSADF II field POSITION operand is derived from the Data Dictionary SEGMENT/WITH/ELEMENT relationship's START attribute.

The IMSADF II TYPE operand is derived from Data Dictionary ELEMENT attributes, whenever possible.

IMSADF II TYPE	DICTIONARY REPRESENTATION
ALPHANUM	TYPE=C
DEC	TYPE=Z
PD	TYPE=P
BIT	TYPE=B
BIN	TYPE=H or F
HEX	TYPE=X or E
FLOAT	TYPE=D
VARCHAR	TYPE=C, SCALE=VARYING, and BYTES=maximum string length

Figure 4-2. IMSADF II TYPE Operand Values

For the IMSADF II operands TYPE=NUM, TYPE=DATE, TYPE=DBCS, and TYPE=MIXED, the ADFDTE01 OTYPE attribute is utilized in addition to ELEMENT attributes.

If the ADFDTE01 OTYPE attribute is NUM or N and

- TYPE attribute of ELEMENT is C (character),

  ADFOUT will produce a field statement with TYPE=NUM.

- TYPE attribute of ELEMENT is Z (zoned decimal) and neither the COBPIC, PLIPIC, nor DECIMALS attribute of ELEMENT indicate a decimal location,

  ADFOUT will produce a FIELD statement with TYPE=NUM.

- TYPE attribute of ELEMENT is Z (zoned decimal) and COBPIC, PLIPIC, or DECIMALS attribute of ELEMENT indicates a decimal location,

  ADFOUT will produce a FIELD statement with TYPE=DEC and a DECIMAL operand and will issue a warning message.

- TYPE attribute of ELEMENT is other than C or Z,

  ADFOUT will produce a FIELD statement with TYPE operand based on the TYPE attribute of ELEMENT and will issue a warning message.

If the ADFDTE01 OTYPE attribute is DATE, DA, or D, and

- TYPE attribute of ELEMENT is C (character) and the BYTES attribute of ELEMENT is 6,

  ADFOUT will produce a field statement with TYPE=DATE.

- TYPE attribute of ELEMENT is Z (zoned decimal), the BYTES attribute of ELEMENT is 6, and neither the COBPIC, PLIPIC, nor DECIMALS attribute of ELEMENT indicate a decimal location,

  ADFOUT will produce a FIELD statement with TYPE=DATE.

- TYPE attribute of ELEMENT is Z (zoned decimal) and COBPIC, PLIPIC, or DECIMALS attribute of ELEMENT indicates a decimal location,

  ADFOUT will produce a FIELD statement with TYPE=DEC and a DECIMAL operand and will issue a warning message.

- TYPE attribute of ELEMENT is C or Z and the BYTES attribute of ELEMENT is other than 6, ADFOUT will produce a FIELD statement with TYPE operand based on the TYPE and BYTES attributes of ELEMENT and will issue a warning message.

- TYPE attribute of ELEMENT is other than C or Z,

  ADFOUT will produce a FIELD statement with TYPE operand based on the TYPE attribute of ELEMENT and will issue a warning message.

If the ADFDTE01 OTYPE attribute is DBCS and

- TYPE attribute of ELEMENT is C (character),

  ADFOUT will produce a FIELD statement with TYPE=DBCS.

- TYPE attribute of ELEMENT is other than C,

  ADFOUT will produce a FIELD statement with TYPE operand based on the TYPE attribute of ELEMENT and will issue a warning message.

If the ADFDTE01 OTYPE attribute is MIXED or M and

- TYPE attribute of ELEMENT is C (character),

  ADFOUT will produce a FIELD statement with TYPE=MIXED.

- TYPE attribute of ELEMENT is other than C,

  ADFOUT will produce a FIELD statement with TYPE operand based on the TYPE attribute of ELEMENT and will issue a warning message.

If the ADEDTE01 OTYPE attribute is ALPHA or A and

- TYPE attribute of ELEMENT is C (character),

  ADFOUT will produce a FIELD statement with TYPE=ALPHA.

- TYPE attribute of ELEMENT is other than C,

  ADFOUT will produce a FIELD statement with TYPE operand based on the TYPE attribute of ELEMENT and will issue a warning message.

The IMSADF II field DECIMAL operand is derived from the Data Dictionary attributes for fields of type Z or P.

- The first attribute checked is COBPIC. If a valid COBPIC is found, it determines the DECIMAL operand. If COBPIC is invalid, a warning message is issued and no DECIMAL operand is generated.

- If there is no COBPIC, PLIPIC is checked. If PLIPIC is valid, it determines the DECIMAL operand. If PLIPIC is invalid, a warning message is issued and no DECIMAL operand is generated.

- If there is no PLIPIC attribute specified, the DECIMALS attribute is checked. If the DECIMALS attribute contains a value between 1 and 13, it is used to generate the DECIMAL operand. If the DECIMALS attribute contains an invalid value, a warning message is issued and no DECIMAL operand is generated. If there is no DECIMALS attribute, no DECIMAL operand is generated.

VALID COBPIC EXAMPLES	VALID PLIPIC EXAMPLES
V999	V99999
	V.99
V9(12)	V(4)9
	V.(10)9

Validity checking on picture clauses begins when an implied decimal point (V) is found.

The IMSADF II field BITOFF operand is derived from the Data Dictionary SEGMENT/WITH/ELEMENT or ELEMENT/WITH/ELEMENT BITSTART relationship attribute.

The IMSADF II generate TRXNAME operand is derived from the first descriptor segment of the IMSADF II segment being processed. The first 30 characters, starting from the first non-blank character of the descriptor segment are used.

If the associated Data Dictionary attribute is blank the ADFOUT processor does not generate the operand. The only exception to this is the IMSADF II field TYPE operand which defaults to ALPHANUM.

ARRAYS

The ADFOUT Processor processes multiple IMSADF II field definitions related to a single standard ELEMENT, if that ELEMENT has been defined to the Data Dictionary as an array. If the ELEMENT is defined with a COBOL OCCURS clause attribute or a PL/I DIMENSION attribute, it is considered an array. If a COBOL OCCURS clause is specified, the first non-blank character specified must be numeric characters specifying the array size. Only the first 5 numeric characters are processed. If a PL/I DIMENSION attribute is specified, the array size is the product of the three dimensions that can be specified.

Once the dimension has been determined, the ADFOUT processor processes up to that number of IMSADF II field definitions related to that ELEMENT. Each IMSADF II field definition is processed as it is encountered. No attempt is made to sequence the IMSADF II fields. Each IMSADF II field definition has the same field length operand. The first IMSADF II field definition has the POSITION operand associated with the SEGMENT/WITH/ELEMENT START relationship attribute. Each subsequent IMSADF II field definition is assigned a POSITION operand of the previous field definition incremented by the LENGTH operand.

Sub elements are supported in single-dimensional arrays such as:

```
01 Array-record.
   03 Array-group-item occurs 10 times.
      03 Subelement-1 PIC X (20)
      03 Subelement-2 PIC X (X).
```

Sub elements are not supported in 2 and 3 dimensional arrays such as:

```
01 Array-record.
   03 Array-item-1 occurs 10 times.
      05 Array-group-item-2 occurs 4 times
         07 Subelement-1 pic X(20)
         07 Subelement-2 pic XX.
```

An example of such an array is:

```
01 Array-record.
   03 Array-item-1 occurs 10 times.
      05 Array-item-2 occurs 5 times.
         07 Array-item-3 Pic S99 occurs 2 times.
```

Array processing ends when there are no additional IMSADF II field definitions to process, or when the number processed equals the array size. If the number of IMSADF II field definitions related to the ELEMENT is less then the dimension, no attempt is made to fill out the segment definition. Even when only one IMSADF II field definition is related to an array ELEMENT, array processing ends after that single definition has been processed. Nested arrays are processed in the same manner. The array size at each level is the product of the dimensions at all levels of nesting including the current level.

Bit string arrays are processed in the same manner.

CHAPTER 5. MESSAGES

This chapter lists all the messages for the IMSADF II Data Dictionary Extension. With each message there is further explanation of the cause of the message, if necessary, plus a brief description of the system action, if any, and the suggested user response, if required.

ABNORMAL TERMINATION CODES

The Data Dictionary Extension does not generate any abnormal termination codes. However, the Dictionary Program Access Facility invokes the ADFOUT processor program which runs within the Data Dictionary environment. Therefore, the Data Dictionary Extension is subject to Dictionary Abnormal Termination Codes. Reference to the manual for OS/VS DB/DC Data Dictionary Terminal User's Guide and Command Reference Dictionary Messages and Codes.

MESSAGE IDENTIFICATION

During Data Dictionary Extension operation you receive printed or displayed messages, some simply informational, and others requiring some action on your part. Each message is preceded by a message identification:

ADFYnnn t

ADF	The component code that distinguishes Data Dictionary Extension messages from other Dictionary messages.
Y	Component code for all IMSADF II Data Dictionary Extension messages.
nnn	Message sequence number
t	Identifies the type of message, as follows:

I	Information message. (Return Code=0)
W	Warning. Execution continues. (Return code=4)
E	Error. Function not executed. Error can be corrected by user. (Return code=8)

You may receive more than one message in a single run or online operation. You receive just one condition code, which corresponds to the most serious type code in the messages issued.

The Data Dictionary Extension messages are primarily divided into two classes: messages associated with parsing and validating the input parameters and messages associated with retrieving data from the Dictionary data bases and creating IMSADF II Rules Generator statements.

The format for each class is unique.

INPUT PARAMETER MESSAGES

The format for almost all input parameter messages is as follows:

ADFYnnn t KEYWORD: Message Text OFFSET-mmm String-value

where:

ADFYnnn t	Message identification, as previously described.
KEYWORD:	There are eleven valid input parameters associated with the Data Dictionary Extension. The same message text can be associated with many of these input parameters. The correct keyword, representing the input parameter that is causing the generated message, will be inserted into the message. The valid keywords are: SYSID: STATUS: OCC: INCLUDE: RULE: TRX: SEG: PGROUP: SOMTX: DEST: IEB:
Message Text	The KEYWORD: is followed by up to 37 characters of message text.
OFFSET-mmm	The constant OFFSET- is followed by the 3-digit offset (mmm) into the parameter string. This is the approximate location within the parameter string where the deviation exists that caused the message to be generated.
String-value	Ten characters of input parameter string data starting from the offset position.

Note: All input parameter messages are treated as error messages (Return code=8). All input errors must be corrected before processing can continue. Reference "Parameters" on page 4-1 for the correct form of all Data Dictionary Extension input parameters.

ADFY001 E KEYWORD: PARAMETER APPEARS TWICE OFFSET-mmm string-value

Explanation: The input parameter appears twice in the parameter string.

System Action: ADFIN/ADFOUT processing is terminated.

Operator Response: Eliminate the duplicate input parameter.

ADFY002 E KEYWORD: PARAMETER TRUNCATED, END of STRING OFFSET-mmm string-value

Explanation: The value associated with the input parameter was truncated by the end of the input parameter string.

System Action: ADFOUT processing is terminated.

Operator Response: Left justify the input parameter string. Parameter string length cannot exceed 200 characters.

ADFY003 E KEYWORD: VALUE LENGTH EXCEEDS 3 CHARACTERS OFFSET-mmm string-value

Explanation: The maximum length of the parameter value is 3 characters. More than 3 characters of data were encountered prior to a delimiter.

System Action: ADFOUT processing is terminated.

Operator Response: Enter a valid parameter value, or insert a delimiter.

ADFY004 E KEYWORD: PARAMETER HAS NO VALUE DEFINED OFFSET-mmm string-value

Explanation: The first character after the keyword is a delimiter, no value was entered for this keyword.

System Action: ADFOUT processing is terminated.

Operator Response: Enter a valid parameter value.

ADFY005 E KEYWORD: VALUE MUST BE Y, N, YES, OR NO OFFSET-mmm string-value

Explanation: A parameter value other than Y, N, YES or NO was encountered.

System Action: ADFOUT processing is terminated.

Operator Response: Enter a valid parameter value.

ADFY006 E KEYWORD: PARAMETER CONTAINS INVALID CHARACTERS OFFSET-mmm string-value

Explanation: Invalid character(s) encountered in the parameter data.

System Action: ADFIN/ADFOUT processing is terminated.

Operator Response: Review valid data characteristics for input parameter, enter a valid parameter value.

ADFY007 E KEYWORD: PARAMETER MISSING OR INVALID OFFSET-mmm string-value

Explanation: End of parameter input reached and a required parameter is missing.

System Action: ADFOUT processing is terminated.

Operator Response: Review required input parameters and enter all required parameters.

ADFY008 E IMBEDDED BLANKS NOT ALLOWED IN STRING OFFSET-mmm string-value

Explanation: A non-blank character was encountered in the parameter string after a blank delimiter character had been found. When a blank delimiter is found, the remainder of the parameter string must be blank.

System Action: ADFIN/ADFOUT processing is terminated.

Operator Response: Change the delimiter to a comma, or remove all non-blank characters after the blank delimiter.

ADFY009 E INVALID DELIMITER AFTER PARAMETER OFFSET-mmm string-value

Explanation: Input parameters must be separated by a valid delimiter, comma. Blanks must follow the last input parameter

System Action: ADFIN/ADFOUT processing is terminated.

Operator Response: Enter a valid delimiter.

ADFY010 E INVALID CHAR IN PARAMETER KEYWORD OFFSET-mmm string-value

Explanation: Input parameter string characters do not match any of the parameter keywords or their valid abbreviations, or the keyword is not followed by an equal sign.

System Action: ADFIN/ADFOUT processing is terminated.

Operator Response: Review input parameter keywords and their valid abbreviations. Enter a valid parameter keyword, followed by an equal sign.

ADFY011 E INCLUDE, RULE, or TRX MUST BE YES

Explanation: The INCLUDE, RULE, and TRX input parameters have all been assigned a value of 'NO', either explicitly or by default. At least one of these parameters must always have a value of 'YES'.

System Action: ADFOUT processing is terminated.

Operator Response: Review input parameter requirements and defaults. Insure that at least one of these parameters has a value of 'YES'.

ADFY012 E COMMA MUST BE FOLLOWED BY A NON-BLANK CHAR OFFSET-mmm string-value

Explanation: A valid delimiter (a comma) was encountered. However, it was followed by a blank character.

System Action: ADFOUT processing is terminated.

Operator Response: Review the input parameter delimiter requirements. If a comma occurs after the last input parameter, change it to a blank, or remove the imbedded blank(s) from the parameter string.

ADFY013 E KEYWORD: KEYWORD FIRST CHAR MUST BE ALPHABETIC OFFSET-mmm string-value

Explanation: The first character of the input parameter keyword must be alphabetic.

System Action: ADFIN/ADFOUT processing is terminated.

Operator Response: Review input parameter keyword restrictions and enter a valid parameter keyword.

ADFY014 E KEYWORD: CODE INVALID, MUST BE A-T, or 0-9 OFFSET-mmm string-value

Explanation: Input parameter value must be A-T, or 0-9.

System Action: ADFOUT processing is terminated.

Operator Response: Review input parameter value restrictions and enter a valid parameter value.

ADFY015 E KEYWORD: VALUE NOT NUMERIC, or EXCEEDS 255 OFFSET-mmm string-value

Explanation: Input parameter value must be a numeric value, and its range is from 0-255.

System Action: ADFOUT processing is terminated.

Operator Response: Review input parameter value restrictions and enter a valid parameter value.

ADFY016 E KEYWORD: PARAMETER - MISSING RIGHT PARENS OFFSET-mmm string-value

Explanation: Input parameter value contains an open 'LEFT' parenthesis and no corresponding close 'RIGHT' parenthesis was found.

System Action: ADFOUT processing is terminated.

Operator Response: Review input parameter value restrictions. Eliminate the open 'left' parenthesis, or add a close 'right' parenthesis.

ADFY017 E KEYWORD: KEYWORD CONTAINS MORE THAN 10 SEGIDS OFFSET-mmm string-value

Explanation: When the list form of the 'SEG=' input parameter is used, it cannot contain more than ten, two-character SEGIDs, each separated by a comma. If more than one SEGID is included in the list, the list must be contained within parenthesis.

System Action: ADFOUT processing is terminated.

Operator Response: Review input parameter value restrictions and enter a valid parameter value.

ADFY018 E KEYWORD: COMMA MUST BE FOLLOWED BY SEGID OFFSET-mmm string-value

Explanation: When the list form of the 'SEG=' input parameter is used and more than one SEGID is entered, each SEGID must be followed by a comma except the last, which must be followed by a closing parenthesis.

System Action: ADFOUT processing is terminated.

Operator Response: Review input parameter value restrictions and enter a valid parameter value.

ADFY019 E KEYWORD: MISSING COMMA BETWEEN SEGIDS OFFSET-mmm string-value

Explanation: When the list form of the 'SEG=' input parameter is used, and more than one SEGID is entered, each SEGID must be delimited from the next SEGID by a comma.

System Action: ADFOUT processing is terminated.

Operator Response: Review input parameter value restrictions and enter a valid delimiter.

ADFY020 E KEYWORD: KEYWORD CONTAINS DUPLICATE SEGIDS OFFSET-mmm string-value

Explanation: When the list form of the 'SEG=' input parameter is used, each two-character SEGID must be unique.

System Action: ADFOUT processing is terminated.

Operator Response: Eliminate duplicate SEGIDs.

ADFY021 E KEYWORD: PARAMETER KEYWORD MUST BE L, P, or T OFFSET-mmm string-value

Explanation: Input parameter value must be L, P, or T.

System Action: ADFOUT processing is terminated.

Operator Response: Review input parameter value restrictions and enter a valid parameter value.

ADFY022 E PASSWORD PARAMETER MUST BE SPECIFIED AND MUST BE NONBLANK WHEN SIGNON PARAMETER IS SPECIFIED

Explanation: A valid SIGNON parameter was entered in the parm list for the ADFIN processor. A valid PASSWORD parameter is required when SIGNON is specified.

System Action: ADFIN processing is terminated.

Operator Response: If your installation uses Data Dictionary security, provide a valid PASSWORD in the ADFIN parm list. If not, remove the SIGNON parm from parm list.

PROCESSING MESSAGES

ADFY029 E KEYWORD: WRITE ERROR

Explanation: A user-specified output file had a write error. The DEST parameter was not specified. Keyword contains the DINCLUDE or DADFOUT keyword, depending on which output file had the error.

System Action: ADFOUT processing is terminated.

Operator Response: Review input and output requirements. Add a valid DINCLUDE or DADFOUT DD statement to the jobstep JCL, or change the output routing by defining the DEST parameter in the input parameter string.

ADFY030 E KEYWORD: IS NOT IN JOBSTEP or DD DUMMY

Explanation: Output was routed to the data set referenced by ddname. This DD statement is either not defined or is specified as DD DUMMY.

System Action: ADFOUT processing is terminated.

Operator Response: Review the output requirements. Add a valid DD statement for the ddname. Alter the output routing by modifying the DEST parameter or the DADFOUT or DINCLUDE parameters or by modifying the INC, RUL, or TRX input parameters.

ADFY031 I PROCESSING SUCCESSFULLY COMPLETED

Explanation: ADFIN/ADFOUT processing was successfully completed with a return code of zero.

System Action: None

Operator Response: None

ADFY032 W PROCESSING SUCCESSFULLY COMPLETED WITH WARNING MESSAGES

Explanation: ADFIN/ADFOUT processing completed successfully but warning messages were generated.

System Action: None

Operator Response: Review the output and determine what actions if any should be taken to eliminate the situation that caused the warnings.

The remainder of the ADFOUT messages are associated with retrieving data from the Dictionary data bases and creating IMSADF II Rules Generator statements.

The format for most of these data base retrieval messages is as follows:

ADFYnnn t Message-text Retrieval-type PARC=mmm Subject-name

where:

ADFYnnn t	Message identification, as previously described.
Message-text	A brief description. In most cases contains the subject category name and the relationship keyword involved in the retrieval request.
Retrieval-type	The type of retrieval requested. Specified as a four-character code. Description of each retrieval type can be found in the DB/DC Data Dictionary Administration and Customization Guide.
PARC=mmm	Return code set by the Data Dictionary upon return from a retrieval request. A description of each return code can be found in the DB/DC Data Dictionary Administration and Customization Guide.
Subject-name	The principle Data Dictionary subject name associated with the retrieval request, in standard Data Dictionary format, 36 characters in length.

ADFY033 E ADFSYS01 RSA PARC=mmm subject-name

Explanation: An attempt has been made to retrieve the ADFSYS01 category attributes using the Dictionary 'RSA' retrieval call, the IMSADF II system extensibility category and the specified subject-name.

System Action: ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error and resubmit the command.

ADFY034 E ADFSYS01/DBS RSRK PARC=mmm subject-name

Explanation: An attempt has been made to retrieve a list of data base subject-names related to the ADFSYS01 subject name using the Dictionary 'RSRK' retrieval call, the IMSADF II system extensibility and the Data base standard categories, and the 'POINT-DBS' relationship keyword.

System Action: ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error and resubmit the command.

ADFY035 E DBS/WITH/SEGMENT RSRK PARC=mmm subject-name

Explanation: An attempt has been made to retrieve a list of segment subject-names related to the data base subject-name using the Dictionary 'RSRK' retrieval call, the standard data base and segment categories, and using the 'WITH' relationship keyword.

System Action: ADFIN/ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error, and resubmit the command.

ADFY036 W DBS HAS NO SEGMENTS PARC=mmm subject-name

Explanation: Dictionary 'RSRK' retrieval, following the standard DBS/WITH/SEGMENT relationship, did not retrieve any segment subject-names.

System Action: Warning message, ADFOUT continues processing.

Operator Response: Review data base relationship to the IMSADF II system being processed.

ADFY037 E SEGMENT/ADFSEG01 RSRK PARC=mmm subject-name

Explanation: An attempt was made to retrieve a list of ADFSEG01 subject-names related to the segment subject name using the Dictionary 'RSRK' retrieval call following the SEGMENT/POINT-ADFSEG/ADFSEG01 relationship.

System Action: ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error and resubmit the command.

ADFY038 W DATABASE HAS NO VALID ADFSEG01

Explanation: All segments related to the given data base have been processed, and no IMSADF II segment statements were generated.

System Action: Warning message, ADFOUT continues processing.

Operator Response: Review data base relationship to the IMSADF II system being processed.

ADFY039 E ADFSEG01 ATTRIBUTES RSA PARC=mmm subject-name

Explanation: An attempt was made to retrieve the ADFSEG01 category attributes using the Dictionary 'RSA' retrieval call, the IMSADF II segment extensibility category, ADFSEG01, and the specified subject-name.

System Action: ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error and resubmit the command.

ADFY040 E SEGMENT HAS NO PARENT PARC=mmm subject-name

Explanation: The DBS/WITH/SEGMENT relationship attributes were retrieved. Both the PARENT and LEVEL attributes are blank.

System Action: ADFOUT processing is terminated.

Operator Response: Insure that all segments have their parent attributes defined in the DBS/WITH/SEGMENT relationship, resubmit the command. The LEVEL attribute is only used for root segments (PARENT=0, or LEVEL=01).

ADFY041 E DBS/WITH/SEGMENT RRA PARC=mmm subject-name

Explanation: An attempt was made to retrieve the DBS/WITH/SEGMENT relationship attributes using the Dictionary 'RRA' retrieval call.

System Action: ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error, and resubmit the command.

ADFY042 E STANDARD SEGMENT RSA PARC=mmm subject-name

Explanation: An attempt was made to retrieve the standard segment category attributes using the Dictionary 'RSA' retrieval call.

System Action: ADFIN/ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error and resubmit the command.

ADFY043 W ADF SEGMENT NAME PARM NOT FOUND - subject-name

Explanation: The IMSADF II segment 'NAME' Rules Generator operand is taken from the Assembler name of the standard segment to which it is related. If the primary name of the standard segment is not Assembler, then an Assembler alias name is used. If no Assembler aliases are found then the first 8 characters of the primary name are used.

System Action: Warning message, ADFOUT continues processing.

Operator Response: Relate an assembler alias to the primary standard segment name if the first 8 characters of the primary name do not match the DBD segment name known to IMS/VS.

ADFY044 E SEGMENT OR DTE ALIAS RA PARC=mmm subject-name

Explanation: Attempting to retrieve standard segment or field aliases using the Dictionary 'RA' retrieval call.

System Action: ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error and resubmit the command.

ADFY045 W ADF SEGMENT KEYNAME ATTRIBUTE - subject-name

Explanation: The IMSADF II segment 'KEYNAME' Rules Generator operand specifies the DBD sequence field name for this segment as it is defined to IMS/VS. When the KEYNAME is not specified as an attribute of the IMSADF II segment it is taken from the assembler name of a standard field that is related to the standard segment currently being processed. All standard fields related to the standard segment are processed until one is found with a sequence indicator attribute of 'Y' or 'M'. If the primary name of this field is not Assembler, then an Assembler alias name is used. If no Assembler aliases are found then the first 8 characters of the primary name are used.

System Action: Warning message, ADFOUT continues processing.

Operator Response: Relate an assembler alias to the primary field name if the first 8 characters of the primary field name do not match the DBD sequence field name.

ADFY046 E SEGMENT HAS NO ELEMENTS PARC=mmm subject-name

Explanation: An attempt was made to retrieve a list of standard elements related to the specified standard segment; no standard elements were found.

System Action: ADFOUT processing is terminated.

Operator Response: Review the Dictionary relationship associated with the specified standard segment subject-name. Either delete the DBS/WITH/SEGMENT relationship or add SEGMENT/WITH/ELEMENT relationships.

ADFY047 E SEGMENT/WITH/DTE RSRK PARC=mmm subject-name

Explanation: An attempt was made to retrieve a list of standard elements related to the specified standard segment subject-name using the Dictionary 'RSRK' retrieval call.

System Action: ADFIN/ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error, and resubmit the command.

ADFY048 E SEGMENT/WITH/DTE RRA PARC=mmm subject-name

Explanation: An attempt was made to retrieve the SEGMENT/WITH/DTE relationship attributes using the Dictionary 'RRA' retrieval call.

System Action: ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error, and resubmit the command.

ADFY049 E DUPLICATE ADF SEGMENT ID - subject-name

Explanation: An IMSADF II segment statement for the IMSADF II SEGID referenced by the specified ADFSEG01 category subject-name has already been processed.

System Action: ADFOUT processing is terminated.

Operator Response: Review ADFSEG01 category subject-name. Each two-character IMSADF II segment ID must be unique within an ADFSEG01 category subject-name. Correct the duplicate name and resubmit the command.

ADFY050 E ADF SEGMENT KEYNAME MISSING - subject-name

Explanation: When the IMSADF II segment KEYNAME parameter is not specified as an attribute in the ADFSEG01 category it is taken from the Assembler name of a related standard field with a sequence indicator attribute of 'U' or 'M'.

System Action: ADFOUT processing is terminated.

Operator Response: Define the IMSADF II segment 'KEYNAME' parameter. Enter the KEYNAME attribute in the related ADFSEG01 category, or define a sequence indicator attribute in a SEGMENT/WITH/DTE relationship. Resubmit the command.

ADFY051 E SEGMENT DESCRIPTOR RDSC PARC=mmm subject-name

Explanation: An attempt was made to retrieve standard segment description text, using the Dictionary 'RDSC' retrieval call.

System Action: ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error and resubmit the command.

ADFY052 E STANDARD SEGMENT PARENT ATTR - subject name

Explanation: Attempting to define the parentage of the specified standard segment subject-name. This information comes from the DBS/WITH/SEGMENT relationship attributes. The physical parent attribute indicates PARENT=0, but the level attribute is not LEVEL=01.

System Action: ADFOUT processing is terminated.

Operator Response: Review DBS/WITH/SEGMENT parent attributes, correct the error and resubmit the command.

ADFY053 E ADF SEGMENT PARENT PARAMETER - subject-name

Explanation: The parent segment of the specified standard segment subject-name does not have a valid ADFSEG01 subject-name related to it. The IMSADF II segment 'PARENT' parameter cannot be established.

System Action: ADFOUT processing is terminated.

Operator Response: Review the standard segment parent attributes in the DBS/WITH/SEGMENT relationship. Review the standard parent segment SEGMENT/POINT-ADFSEG/ADFSEG01 relationships to the ADFSEG01 category. Correct the error and resubmit the command.

ADFY054 E ADFSEG01 - PARENT RSRK PARC=mmm subject-name

Explanation: Attempting to retrieve a list of ADFSEG01 category subject-names related to the specified standard segment parent subject-name using the Dictionary 'RSRK' retrieval call.

System Action: ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error and resubmit the command.

ADFY055 E ADF PARENT ATTR RSA PARC=mmm subject-name

Explanation: An attempt was made to retrieve IMSADF II parent segment attributes from the ADFSEG01 category attributes using the Dictionary 'RSA' retrieval call.

System Action: ADFOUT processing is terminated

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error and resubmit the command.

ADFY056 W NO DICT. ENTRY FOUND FOR INPUT SEGID, REVIEW OUTPUT FOR MISSING IDs

Explanation: The 'SEG=' input parameter contains a list of two-character IMSADF II SEGIDS. All IMSADF II segments that are related to the specified IMSADF II SYSTEM ID were processed and at least one of the input IMSADF II SEGIDS was not found.

System Action: ADFOUT processing completed.

Operator Response: Review input list of SEGIDS with output, to find which SEGIDS were not processed. Change input value or add new IMSADF II SEGMENT to dictionary, resubmit the command.

ADFY057 E NO ADF FIELDS PROCESSED PARC=mmm subject-name

Explanation: All standard elements that are related to the specified standard segment subject-name were processed and no-related ADFDTE01 elements were found that qualified as valid IMSADF II elements related to the current IMSADF II system and IMSADF II segment being processed.

System Action: ADFOUT processing is terminated.

Operator Response: Review the ADFDTE01 category subject-name naming conventions defined in "Additional Extensibility Categories" on page 2-2. Each IMSADF II field must be unique to an IMSADF II segment and/or IMSADF II system. Correct the errors, and resubmit the command.

ADFY058 E DTE/POINT/ADFDTE01 RSRK PARC=mmm subject-name

Explanation: An attempt was made to retrieve a list of ADFDTE01 category subject-names related to the specified standard element subject-name using the Dictionary 'RSRK' retrieval call.

System Action: ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error and resubmit the command.

ADFY059 E ADF DTE ATTRIBUTES RSA PARC=mmm subject-name

Explanation: An attempt was made to retrieve ADFDTE01 category attributes for the specified subject-name using the Dictionary 'RSA' retrieval call.

System Action: ADFIN/ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error and resubmit the command.

ADFY060 E STANDARD DTE ATTRS RSA PARC=mmm subject-name

Explanation: An attempt was made to retrieve standard element category attributes for the specified subject-name using the Dictionary 'RSA' retrieval call.

System Action: ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error and resubmit the command.

ADFY062 W UNKNOWN DATA TYPE, DEFAULT TAKEN subject-name

Explanation: The data type attribute for the specified standard element subject-name is invalid to both the Dictionary and IMSADF II. No IMSADF II field 'TYPE' parameter will be generated for this field. The IMSADF II default of TYPE=ALPHANUM will be in effect.

System Action: Warning message, ADFOUT processing continues.

Operator Response: Correct the invalid data type attribute and resubmit the command.

ADFY063 W INVALID BITSTART RANGE, IGNORED subject-name

Explanation: The BITSTART attribute for the specified standard element subject-name is outside the valid range of 1-8. The IMSADF II field 'BITOFF' parameter will not be generated.

System Action: Warning message, ADFOUT processing continues.

Operator Response: Correct the invalid BITSTART attribute, and resubmit the command.

ADFY064 W BITSTART SPECIFIED, DATA TYPE NOT BIT subject-name

Explanation: The BITSTART attribute for the specified standard element subject-name is ignored even though it has a value in it. The DATA TYPE attribute must be BIT before the BITSTART attribute is processed.

System Action: Warning message, ADFOUT processing continues.

Operator Response: Verify the DATA TYPE and BITSTART attributes for the specified standard element subject-name. Correct the error, and resubmit the command.

ADFY065 E ARRAY ELEMENT CONTAINS NO SUB-ELEMENTS subject-name

Explanation: The specified standard element subject-name was previously marked as an array element containing sub-elements. An attempt to retrieve a list of these sub-element subject-name failed. Potential ADFOUT processing logic error.

System Action: ADFOUT processing is terminated.

Operator Response: Review specified standard element attributes and relationships. If no discrepancies are found and error persists, notify your IBM representative.

ADFY066 E DTE/CONTAINS/DTE RSRK PARC=mmm subject-name

Explanation: An attempt was made to retrieve a list of sub-element subject-names related to the specified standard element subject-name using the Dictionary 'RSRK' retrieval call.

System Action: ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error and resubmit the command.

ADFY067 E DTE/CONTAINS/DTE RRA PARC=mmm subject-name

Explanation: An attempt was made to retrieve the DTE/CONTAINS/DTE relationship attributes using the Dictionary 'RRA' retrieval call.

System Action: ADFOUT processing is terminated.

Operator Response: Evaluate the Dictionary return code 'PARC=mmm', correct the error and resubmit the command.

ADFY068 E ELEMENT ALREADY IN ARRAY TABLE subject-name

Explanation: The specified standard element subject-name was previously marked as an array element containing sub-elements. The same standard element should not be processed twice within the same segment. Potential ADFOUT processing logic error.

System Action: ADFOUT processing is terminated.

Operator Response: Review specified standard element relationships. If no discrepancies are found and error persists, notify your IBM representative.

ADFY069 E NO DICTIONARY ENTRY FOUND FOR ANY OF THE INPUT ADF SEGIDS

Explanation: All IMSADF II segments within the specified IMSADF II system ID have been searched. None of these segments match the IMSADF II segment IDs contained in the SEG parameter of the Data Dictionary EXECUTE command.

System Action: ADFOUT processing is terminated.

Operator Response: Review the IMSADF II data base structure. Specify a valid IMSADF II system ID (SYSID) and segment ID (SEGID) combination and resubmit the command.

ADFY070 W DEC ATTR, INVALID PIC

Explanation: An invalid value was encountered in the Data Dictionary COBPIC or PLIPIC operand. No DECIMAL attribute is calculated.

System Action: Warning message, ADFOUT processing continues.

Operator Response: Correct the invalid COBPIC or PLIPIC operand and resubmit the command.

ADFY071 W DEC ATTR, INVALID DECIMALS

Explanation: No COBPIC or PLIPIC operand was found; therefore, the Data Dictionary DECIMALS attribute was checked. The DECIMALS attribute field contained a value other than a positive numeric value between 1 and 13, the range that is allowed by IMSADF II. No DECIMAL operand calculated.

System Action: Warning message, ADFOUT processing continues.

Operator Response: Correct the invalid Data Dictionary DECIMALS attribute or add a valid COBPIC or PLIPIC value and resubmit the command.

ADFY072 W KEY=YES OMITTED IN ADF SEG seg-id

Explanation: No IMSADF II field statement with KEY=YES was generated. This would result in a Rules Generator error if the output was sent to the Rules Generator without modification.

System Action: Warning message, ADFOUT processing continues.

Operator Response: Review the ADFDTE01 subjects related to the ADFSEG01 and specify the appropriate ADFDTE01 subject's key attribute as YES.

ADFY073 W ADFDTE01 OTYPE, DTE TYPE CONFLICT

Explanation: An OTYPE was specified for the ADFDTE01 that is not consistent with the TYPE specified for the related standard ELEMENT.

System Action: Warning message, ADFOUT processing continues.

Operator Response: Review "Additional Operands" on page 4-8 of Chapter 4, "ADFOUT Processor" and determine the correct usage of OTYPE for this situation.

ADFY074 W ADFDTE01 OTYPE, DTE DEC CONFLICT

Explanation: An OTYPE was specified for the ADFDTE01 that is not consistent with the DECIMAL attribute of the standard ELEMENT.

System Action: Warning message, ADFOUT processing continues.

Operator Response: Review "Additional Operands" on page 4-8 of Chapter 4, "ADFOUT Processor" and determine the correct usage of OTYPE for this situation.

ADFY075 W ADFDTE01 OTYPE, DTE LENGTH CONFLICT

Explanation: An OTYPE was specified for the ADFDTE01 that is not consistent with the length specified for the related standard ELEMENT.

System Action: Warning message, ADFOUT processing continues.

Operator Response: Review "Additional Operands" on page 4-8 of Chapter 4, "ADFOUT Processor" and determine the correct usage of OTYPE for this situation.

ADFY076 E DDADFX DDNAME NOT SPECIFIED IN JCL

Explanation: The ddname associated with the ADFX Extract file was not present in the JCL used to execute the ADFIN processor. If the DDADFX parameter was not specified in the ADFIN parm list, the default ddname is DDADFX.

System Action: ADFIN processing is terminated.

Operator Response: Add the necessary DD statement to the JCL for the ADFX Extract file.

ADFY077 E DDADFCMD DDNAME NOT SPECIFIED IN JCL

Explanation: The ddname associated with the ADFCMD Command file was not present in the JCL used to execute the ADFIN processor. If the DDADFCMD parameter was not specified in the ADFIN parm list, the default ddname is DDADFCMD.

System Action: ADFIN processing is terminated.

Operator Response: Add the necessary DD statement to the JCL for the ADFCMD Command file.

ADFY078 E DDADFX FIRST RECORD INVALID, RECTYPE NOT EQUAL TO ADFSYS

Explanation: The first record in the ADFX Extract file contains invalid data.

System Action: ADFIN processing is terminated.

Operator Response: Check the JCL to ensure that the DD statement for the DDADFX Extract file refers to the data set created by the Rules Generator Extract module (with LRECL 256). Do not confuse this file with the ISPF Extract file also created by the Rules Generator Extract module (with LRECL 1200). If necessary, resubmit the Rules Generator job.

ADFY079 E ERROR IN RETRIEVING PSB SUBJECT WITH SUBJECTNAME = subject-name AND RETURN CODE mmm

Explanation: An attempt was made to retrieve the PSB subject using the Dictionary 'RSA' retrieval call.

System Action: ADFIN processing is terminated.

Operator Response: Evaluate the Dictionary return code (mmm), correct the error and resubmit the ADFIN job.

ADFY080 E ERROR IN RETRIEVING PCB SUBJECT WITH SUBJECTNAME = subject-name AND RETURN CODE mmm

Explanation: An attempt was made to retrieve the PCB subject using the Dictionary 'RSA' retrieval call.

System Action: ADFIN processing is terminated.

Operator Response: Evaluate the Dictionary return code (mmm), correct the error and resubmit the ADFIN job.

ADFY081 E ERROR IN RETRIEVING DBS SUBJECT WITH SUBJECTNAME = subject-name AND RETURN CODE mmm

Explanation: An attempt was made to retrieve the DBS subject using the Dictionary 'RSA' retrieval call.

System Action: ADFIN processing is terminated.

Operator Response: Evaluate the Dictionary return code (mmm), correct the error and resubmit the ADFIN job.

ADFY082 E ADFSYS01 SUBJECT NOT PRESENT IN DICTIONARY AND UPDATE = NO SUBJECTNAME = subject-name

Explanation: The UPDATE parm in the ADFIN parm list indicated that the user did not want the ADFSYS01 subject to be updated. However, the ADFSYS01 subject does not exist in the Dictionary.

System Action: ADFIN processing is terminated.

Operator Response: There are two choices: Use a Dictionary ADD command to add the ADFSYS01 subject, or change the value of the UPDATE parameter in the ADFIN parm list to allow ADFIN to create the ADD command. Then, resubmit the ADFIN job.

ADFY083 E ERROR IN RETRIEVING ADFSYS01 SUBJECT WITH SUBJECTNAME = subject-name AND RETURN CODE mmm

Explanation: An attempt was made to retrieve the ADFSYS01 subject using the Dictionary 'RSA' retrieval call.

System Action: ADFIN processing is terminated.

Operator Response: Evaluate the Dictionary return code (mmm), correct the error and resubmit the ADFIN job.

ADFY084 W ERROR IN RETRIEVING ADFSEG01 SUBJECT WITH SUBJECTNAME = subject-name AND RETURN CODE mmm

Explanation: An attempt was made to retrieve the ADFSEG01 subject using the Dictionary 'RSA' retrieval call.

System Action: ADFIN processing of the current segment is terminated.

Operator Response: Evaluate the Dictionary return code (mmm), correct the error and resubmit the ADFIN job.

ADFY085 W ERROR IN RETRIEVING STANDARD SEGMENT SUBJECT WITH SUBJECTNAME = subject-name AND RETURN CODE mmm

Explanation: An attempt was made to retrieve the SEGMENT subject using the Dictionary 'RSA' retrieval call.

System Action: ADFIN processing of the current segment is terminated.

Operator Response: Evaluate the Dictionary return code (mmm), correct the error and resubmit the ADFIN job.

ADFY086 W ERROR IN RETRIEVING ADFSEG01 SUBJECT RELATED TO ADFSYS01 SUBJECT SUBJECTNAME = subject-name, RETURN CODE mmm

Explanation: An attempt was made to retrieve the ADFSEG01 subject using the Dictionary 'RSRK' retrieval call.

System Action: ADFIN processing of the current segment is terminated.

Operator Response: Evaluate the Dictionary return code (mmm), correct the error and resubmit the ADFIN job.

ADFY087 W ADFSEG01 SUBJECT NOT PRESENT IN DICTIONARY AND UPDATE = NO SUBJECTNAME = subject-name

Explanation: The UPDATE parm in the ADFIN parm list indicated that the user did not want the ADFSEG01 subject to be updated. However, the ADFSEG01 subject does not exist in the Dictionary.

System Action: ADFIN processing of the current segment is terminated.

Operator Response: There are two choices: Use a Dictionary ADD command to add the ADFSEG01 subject, or change the value of the UPDATE parameter in the ADFIN parm list to allow ADFIN to create the ADD command. Then, resubmit the ADFIN job.

ADFY088 E DDADFX FILE CONTAINS ADFDTE RECORD BEFORE ADFSEG RECORD

Explanation: The ADFX Extract file contains an ADFDTE record before any ADFSEG records.

System Action: ADFIN processing is terminated.

Operator Response: Check the JCL to ensure that the DD statement for the DDADFX Extract file refers to the data set created by the Rules Generator Extract module (with LRECL 256). Do not confuse this file with the ISPF Extract file also created by the Rules Generator Extract module (with LRECL 1200). If necessary, resubmit the Rules Generator job.

ADFY089 W ERROR IN RETRIEVING ADFDTE01 SUBJECT RELATED TO ADFSEG01 SUBJECT WITH SUBJECTNAME = subject-name, RETURN CODE mmm

Explanation: An attempt was made to retrieve the ADFDTE01 subject using the Dictionary 'RSRK' retrieval call.

System Action: ADFIN processing of the current segment is terminated.

Operator Response: Evaluate the Dictionary return code (mmm), correct the error and resubmit the ADFIN job.

ADFY090 W INVALID DATA IN DICTIONARY ADFDTE01 SUBJECT SYSID AND/OR SEGID FIELDS, SUBJECTNAME = subject-name

Explanation: The ADFDTE01 subject SYSID and/or SEGID attributes are invalid. If a concatenated username is used, the SYSID and SEGID fields must be blank.

System Action: ADFIN processing of the current segment is terminated.

Operator Response: Use the Dictionary CHANGE_IN command to blank out the SYSID and SEGID attributes of the ADFDTE01 subject. Then, resubmit the ADFIN job.

ADFY091 W ADFDTE01 SUBJECT NOT PRESENT IN DICTIONARY AND UPDATE = NO SUBJECTNAME = subject-name

Explanation: The UPDATE parm in the ADFIN parm list indicated that the user did not want the ADFDTE01 subject to be updated. However, the ADFDTE01 subject does not exist in the Dictionary.

System Action: ADFIN processing of the current segment is terminated.

Operator Response: There are two choices: Use a Dictionary ADD command to add the ADFDTE01 subject, or change the value of the UPDATE parameter in the ADFIN parm list to allow ADFIN to create the ADD command. Then, resubmit the ADFIN job.

ADFY092 W STANDARD ELEMENT SUBJECT CORRESPONDING TO ADFDTE01 SUBJECT (SUBJECTNAME subject-name) NOT PRESENT IN DICTIONARY

Explanation: The standard ELEMENT subject corresponding to the ADFDTE01 subject shown was not present in the Dictionary.

System Action: ADFIN processing of the current segment is terminated.

Operator Response: Use the Dictionary ADD command to add the standard ELEMENT to the Dictionary. Use the Dictionary ADD_RELATIONSHIP command to relate this ELEMENT subject to the standard SEGMENT subject. Then, resubmit the ADFIN job.

ADFY093 W ERROR IN RETRIEVING STANDARD ELEMENT SUBJECT WITH SUBJECTNAME = subject-name AND RETURN CODE mmm

Explanation: An attempt was made to retrieve the ELEMENT subject using the Dictionary 'RSA' retrieval call.

System Action: ADFIN processing of the current segment is terminated.

Operator Response: Evaluate the Dictionary return code (mmm), correct the error and resubmit the ADFIN job.

ADFY094 W ERROR IN RETRIEVING STANDARD SUBELEMENT SUBJECT WITH SUBJECTNAME = subject-name AND RETURN CODE mmm

Explanation: An attempt was made to retrieve an ELEMENT subject related to another ELEMENT subject using the Dictionary 'RSRK' retrieval call.

System Action: ADFIN processing of the current segment is terminated.

Operator Response: Evaluate the Dictionary return code (mmm), correct the error and resubmit the ADFIN job.

INSTALLATION

Before the installation of the Data Dictionary Extension of IMSADF II, the DB/DC Data Dictionary, Release 5 or later, must be installed.

Two mutually exclusive installation procedures are provided. Which procedure to use is determined by whether the ADFOUT or IMSADF II, Release 1, Version 1, is installed.

One installation procedure contains a Dictionary command stream to modify the model installed as part of IMSADF II, Release 1, and is contained in IMSADF.JCLLIB (DDEECI5). The other procedure is used when the model from IMSADF II, Release 1, Version 1, is not installed and is in IMSADF.JCLLIB (DDEECI).

APPENDIX A. ADFIN PROCESSOR SAMPLE PROCEDURE OUTPUT

```
FLUSH=NO;
********************************************************************************
*
* ADFIN - DATA DICTIONARY INPUT COMMANDS DERIVED FROM DATA EXTRACTED FROM ADF
*
********************************************************************************
*
*        ADFSYS01:  SAMP
*
********************************************************************************
A        ADFSYS01 (T, ,SAMP,000)                                               ;
CI       ADFSYS01 (T, ,SAMP,000)                             TO                +
         SOMTX=OR                                                              +
         PGROUP=ZZ                                                             +
         PCBNO=001                                                             +
         DBID=PA                                                               +
         ADFID=MFC1                                                            +
         STRAILER=1                                                            +
         SHEADING='S A M P L E   P R O B L E M'                                +
         SFORMAT=DASH                                                          ;
AR       ADFSYS01 (T, ,SAMP,000)                             POINT_DBS         +
         DATABASE (T,P,DI21PART,000)                                           ;
********************************************************************************
*
*        ADFSEG01:  PA
*        SEGMENT :  PARTROOT
*
********************************************************************************
A        ADFSEG01 (T, ,SAMPPA,000)                                             ;
CI       ADFSEG01 (T, ,SAMPPA,000)                           TO                +
         PCBNO=001                                                             +
         DBID=PA                                                               +
         TRAILER=01                                                            +
         SKSEGS=18                                                             +
         KASCEND=YES                                                           +
         ADBSNAME=DI21PART                                                     +
         ALENGTH=00050                                                         ;
AR       ADFSYS01 (T, ,SAMP,000)                             ADFSYS_HAS        +
         ADFSEG01 (T, ,SAMPPA,000)                                             ;
AR       ADFSEG01 (T, ,SAMPPA,000)                           POINT_SEG         +
         SEGMENT  (T,A,PARTROOT,000)                                           ;
*
*        ADFDTE01:  SAPAKEY
*        ELEMENT :  PART-NUMBER
*
A        ADFDTE01 (T, ,SAPAKEY,000)                                            ;
CI       ADFDTE01 (T, ,SAPAKEY,000)                          TO                +
         KEY=YES                                                               +
         SIGN=NO                                                               +
         SNAME='PART NUMBER'                                                   +
         RELATED=NO                                                            +
         AUDIT=NO                                                              +
         CAUDIT=NO                                                             +
         MSG=NO                                                                ;
AR       ADFSEG01 (T, ,SAMPPA,000)                           ADFSEG_HAS        +
         ADFDTE01 (T, ,SAPAKEY,000)                                            ;
AR       ADFDTE01 (T, ,SAPAKEY,000)                          POINT_DTE         +
         ELEMENT  (T,C,PART-NUMBER,000)                                        ;
*
*        ADFDTE01:  SAPADESC
*        ELEMENT :  PART-DESC
*
A        ADFDTE01 (T, ,SAPADESC,000)                                           ;
CI       ADFDTE01 (T, ,SAPADESC,000)                         TO                +
         KEY=NO                                                                +
         SIGN=NO                                                               +
         SNAME='DESCRIPTION'                                                   +
         RELATED=YES                                                           +
         AUDIT=NO                                                     +
         CAUDIT=NO                                                    +
         MSG=NO                                                       ;
AR       ADFSEG01 (T, ,SAMPPA,000)                     ADFSEG_HAS     +
         ADFDTE01 (T, ,SAPADESC,000)                                  ;
AR       ADFDTE01 (T, ,SAPADESC,000)                   POINT_DTE      +
         ELEMENT  (T,C,PART-DESC,000)                                 ;
************************************************************************
*
*        ADFSEG01:  PD
*        SEGMENT :  STANINFO
*
************************************************************************
A        ADFSEG01 (T, ,SAMPPD,000)                                    ;
CI       ADFSEG01 (T, ,SAMPPD,000)                     TO             +
         PCBNO=001                                                    +
         DBID=PA                                                      +
         TRAILER=01                                                   +
         SKSEGS=37                                                    +
         KASCEND=YES                                                  +
         ADBSNAME=DI21PART                                            +
         ALENGTH=00085                                                ;
AR       ADFSYS01 (T, ,SAMP,000)                       ADFSYS_HAS     +
         ADFSEG01 (T, ,SAMPPD,000)                                    ;
AR       ADFSEG01 (T, ,SAMPPD,000)                     POINT_SEG      +
         SEGMENT  (T,A,STANINFO,000)                                  ;
*
*        ADFDTE01:  SAPDKEY
*        ELEMENT :  STANKEY
*
A        ADFDTE01 (T, ,SAPDKEY,000)                                   ;
CI       ADFDTE01 (T, ,SAPDKEY,000)                    TO             +
         KEY=YES                                                      +
         SIGN=NO                                                      +
         SNAME='KEY FIELD'                                            +
         RELATED=NO                                                   +
         AUDIT=NO                                                     +
         CAUDIT=NO                                                    +
         MSG=NO                                                       ;
AR       ADFSEG01 (T, ,SAMPPD,000)                     ADFSEG_HAS     +
         ADFDTE01 (T, ,SAPDKEY,000)                                   ;
AR       ADFDTE01 (T, ,SAPDKEY,000)                    POINT_DTE      +
         ELEMENT  (T,C,STANKEY,000)                                   ;
*
*        ADFDTE01:  SAPDPRCD
*        ELEMENT :  PROC-CODE
*
A        ADFDTE01 (T, ,SAPDPRCD,000)                                  ;
CI       ADFDTE01 (T, ,SAPDPRCD,000)                   TO             +
         KEY=NO                                                       +
         SIGN=NO                                                      +
         SNAME='PROC CODE'                                            +
         RELATED=NO                                                   +
         AUDIT=NO                                                     +
         CAUDIT=NO                                                    +
         MSG=NO                                                       ;
AR       ADFSEG01 (T, ,SAMPPD,000)                     ADFSEG_HAS     +
         ADFDTE01 (T, ,SAPDPRCD,000)                                  ;
AR       ADFDTE01 (T, ,SAPDPRCD,000)                   POINT_DTE      +
         ELEMENT  (T,C,PROC-CODE,000)                                 ;
*
*        ADFDTE01:  SAPDINVC
*        ELEMENT :  INV-CODE
*
A        ADFDTE01 (T, ,SAPDINVC,000)                                  ;
CI       ADFDTE01 (T, ,SAPDINVC,000)                   TO             +
         KEY=NO                                                       +
         SIGN=NO                                                      +
         SNAME='INVENTORY CODE'                                       +
         RELATED=NO                                                   +
         AUDIT=NO                                                     +
         CAUDIT=NO                                                    +
         MSG=NO                                                       ;
AR       ADFSEG01 (T, ,SAMPPD,000)                     ADFSEG_HAS     +
        ADFDTE01 (T, ,SAPDINVC,000)                                     ;
AR      ADFDTE01 (T, ,SAPDINVC,000)                 POINT_DTE           +
        ELEMENT  (T,C,INV-CODE,000)                                     ;
*
*       ADFDTE01:  SAPDPLRV
*       ELEMENT :  PLAN-REV-NO
*
A       ADFDTE01 (T, ,SAPDPLRV,000)                                     ;
CI      ADFDTE01 (T, ,SAPDPLRV,000)                 TO                  +
        KEY=NO                                                          +
        SIGN=NO                                                         +
        SNAME='PLAN REV NO'                                             +
        RELATED=NO                                                      +
        AUDIT=NO                                                        +
        CAUDIT=NO                                                       +
        MSG=NO                                                          ;
AR      ADFSEG01 (T, ,SAMPPD,000)                   ADFSEG_HAS          +
        ADFDTE01 (T, ,SAPDPLRV,000)                                     ;
AR      ADFDTE01 (T, ,SAPDPLRV,000)                 POINT_DTE           +
        ELEMENT  (T,C,PLAN-REV-NO,000)                                  ;
*
*       ADFDTE01:  SAPDMKDP
*       ELEMENT :  MAKE-DEPT
*
A       ADFDTE01 (T, ,SAPDMKDP,000)                                     ;
CI      ADFDTE01 (T, ,SAPDMKDP,000)                 TO                  +
        KEY=NO                                                          +
        SIGN=NO                                                         +
        SNAME='MAKE DEPT'                                               +
        RELATED=NO                                                      +
        AUDIT=NO                                                        +
        CAUDIT=NO                                                       +
        MSG=NO                                                          ;
AR      ADFSEG01 (T, ,SAMPPD,000)                   ADFSEG_HAS          +
        ADFDTE01 (T, ,SAPDMKDP,000)                                     ;
AR      ADFDTE01 (T, ,SAPDMKDP,000)                 POINT_DTE           +
        ELEMENT  (T,C,MAKE-DEPT,000)                                    ;
*
*       ADFDTE01:  SAPDCOMM
*       ELEMENT :  COMM-CODE
*
A       ADFDTE01 (T, ,SAPDCOMM,000)                                     ;
CI      ADFDTE01 (T, ,SAPDCOMM,000)                 TO                  +
        KEY=NO                                                          +
        SIGN=NO                                                         +
        SNAME='COMM CODE'                                               +
        RELATED=NO                                                      +
        AUDIT=NO                                                        +
        CAUDIT=NO                                                       +
        MSG=NO                                                          ;
AR      ADFSEG01 (T, ,SAMPPD,000)                   ADFSEG_HAS          +
        ADFDTE01 (T, ,SAPDCOMM,000)                                     ;
AR      ADFDTE01 (T, ,SAPDCOMM,000)                 POINT_DTE           +
        ELEMENT  (T,C,COMM-CODE,000)                                    ;
*
*       ADFDTE01:  SAPDRISP
*       ELEMENT :  RIGHT-MAKE-TIME
*
A       ADFDTE01 (T, ,SAPDRISP,000)                                     ;
CI      ADFDTE01 (T, ,SAPDRISP,000)                 TO                  +
        KEY=NO                                                          +
        SIGN=YES                                                        +
        SNAME='RIGHT MAKE TIME'                                         +
        RELATED=NO                                                      +
        AUDIT=NO                                                        +
        CAUDIT=NO                                                       +
        MSG=NO                                                          ;
AR      ADFSEG01 (T, ,SAMPPD,000)                   ADFSEG_HAS          +
        ADFDTE01 (T, ,SAPDRISP,000)                                     ;
AR      ADFDTE01 (T, ,SAPDRISP,000)                 POINT_DTE           +
        ELEMENT  (T,C,RIGHT-MAKE-TIME,000)                              ;
*
*       ADFDTE01:  SAPDWRSP
*       ELEMENT :  WRONG-MAKE-TIME
```

```
*
A        ADFDTE01 (T, ,SAPDWRSP,000)                                       ;
CI       ADFDTE01 (T, ,SAPDWRSP,000)                   TO                  +
         KEY=NO                                                            +
         SIGN=YES                                                          +
         SNAME='WRONG MAKE TIME'                                           +
         RELATED=NO                                                        +
         AUDIT=NO                                                          +
         CAUDIT=NO                                                         +
         MSG=NO                                                            ;
AR       ADFSEG01 (T, ,SAMPPD,000)                     ADFSEG_HAS          +
         ADFDTE01 (T, ,SAPDWRSP,000)                                       ;
AR       ADFDTE01 (T, ,SAPDWRSP,000)                   POINT_DTE           +
         ELEMENT  (T,C,WRONG-MAKE-TIME,000)                                ;
***************************************************************************
*
*        ADFSEG01:  IV
*        SEGMENT :  STOKSTAT
*
***************************************************************************
A        ADFSEG01 (T, ,SAMPIV,000)                                         ;
CI       ADFSEG01 (T, ,SAMPIV,000)                     TO                  +
         PCBNO=001                                                         +
         DBID=PA                                                           +
         TRAILER=01                                                        +
         SKLEFT(1)='INVENTORY         UNIT      CURRENT'                   +
         SKLEFT(2)='LOCATION          PRICE     REQMNTS'                   +
         SKRIGHT(1)='   ON      TOTAL      DISBURSEMENTS'                  +
         SKRIGHT(2)=' ORDER    STOCK   PLANNED  UNPLANNED'                 +
         SKSEGS=37                                                         +
         KASCEND=YES                                                       +
         ADBSNAME=DI21PART                                                 +
         AKEYNAME=STOCKEY                                                  +
         ALENGTH=00160                                                     ;
AR       ADFSYS01 (T, ,SAMP,000)                       ADFSYS_HAS          +
         ADFSEG01 (T, ,SAMPIV,000)                                         ;
AR       ADFSEG01 (T, ,SAMPIV,000)                     POINT_SEG           +
         SEGMENT  (T,A,STOKSTAT,000)                                       ;
*
*        ADFDTE01:  SAIVW
*        ELEMENT :  FILL-0
*
A        ADFDTE01 (T, ,SAIVW,000)                                          ;
CI       ADFDTE01 (T, ,SAIVW,000)                      TO                  +
         KEY=YES                                                           +
         SIGN=NO                                                           +
         SNAME='00'                                                        +
         RELATED=NO                                                        +
         COLUMN=01                                                         +
         AUDIT=NO                                                          +
         CAUDIT=NO                                                         +
         MSG=NO                                                            ;
AR       ADFSEG01 (T, ,SAMPIV,000)                     ADFSEG_HAS          +
         ADFDTE01 (T, ,SAIVW,000)                                          ;
AR       ADFDTE01 (T, ,SAIVW,000)                      POINT_DTE           +
         ELEMENT  (T,C,FILL-0,000)                                         ;
*
*        ADFDTE01:  SAIVAREA
*        ELEMENT :  AREA
*
A        ADFDTE01 (T, ,SAIVAREA,000)                                       ;
CI       ADFDTE01 (T, ,SAIVAREA,000)                   TO                  +
         KEY=YES                                                           +
         SIGN=NO                                                           +
         SNAME='AREA'                                                      +
         RELATED=NO                                                        +
         COLUMN=03                                                         +
         AUDIT=NO                                                          +
         CAUDIT=NO                                                         +
         MSG=NO                                                            ;
AR       ADFSEG01 (T, ,SAMPIV,000)                     ADFSEG_HAS          +
         ADFDTE01 (T, ,SAIVAREA,000)                                       ;
AR       ADFDTE01 (T, ,SAIVAREA,000)                   POINT_DTE           +
         ELEMENT  (T,C,AREA,000)                                               ;
*
*        ADFDTE01:  SAIVINVD
*        ELEMENT :  INV-DEPT
*
A        ADFDTE01 (T, ,SAIVINVD,000)                                           ;
CI       ADFDTE01 (T, ,SAIVINVD,000)                         TO                +
         KEY=YES                                                               +
         SIGN=NO                                                               +
         SNAME='INV DEPT'                                                      +
         RELATED=NO                                                            +
         COLUMN=04                                                             +
         AUDIT=NO                                                              +
         CAUDIT=NO                                                             +
         MSG=NO                                                                ;
AR       ADFSEG01 (T, ,SAMPIV,000)                           ADFSEG_HAS        +
         ADFDTE01 (T, ,SAIVINVD,000)                                           ;
AR       ADFDTE01 (T, ,SAIVINVD,000)                         POINT_DTE         +
         ELEMENT  (T,C,INV-DEPT,000)                                           ;
*
*        ADFDTE01:  SAIVPROJ
*        ELEMENT :  PROJECT
*
A        ADFDTE01 (T, ,SAIVPROJ,000)                                           ;
CI       ADFDTE01 (T, ,SAIVPROJ,000)                         TO                +
         KEY=YES                                                               +
         SIGN=NO                                                               +
         SNAME='PROJECT'                                                       +
         RELATED=NO                                                            +
         COLUMN=06                                                             +
         AUDIT=NO                                                              +
         CAUDIT=NO                                                             +
         MSG=NO                                                                ;
AR       ADFSEG01 (T, ,SAMPIV,000)                           ADFSEG_HAS        +
         ADFDTE01 (T, ,SAIVPROJ,000)                                           ;
AR       ADFDTE01 (T, ,SAIVPROJ,000)                         POINT_DTE         +
         ELEMENT  (T,C,PROJECT,000)                                            ;
*
*        ADFDTE01:  SAIVDIV
*        ELEMENT :  DIVISION
*
A        ADFDTE01 (T, ,SAIVDIV,000)                                            ;
CI       ADFDTE01 (T, ,SAIVDIV,000)                          TO                +
         KEY=YES                                                               +
         SIGN=NO                                                               +
         SNAME='DIVISION'                                                      +
         RELATED=NO                                                            +
         COLUMN=9                                                              +
         AUDIT=NO                                                              +
         CAUDIT=NO                                                             +
         MSG=NO                                                                ;
AR       ADFSEG01 (T, ,SAMPIV,000)                           ADFSEG_HAS        +
         ADFDTE01 (T, ,SAIVDIV,000)                                            ;
AR       ADFDTE01 (T, ,SAIVDIV,000)                          POINT_DTE         +
         ELEMENT  (T,C,DIVISION,000)                                           ;
*
*        ADFDTE01:  SAIVFILL
*        ELEMENT :  DIV-FILL
*
A        ADFDTE01 (T, ,SAIVFILL,000)                                           ;
CI       ADFDTE01 (T, ,SAIVFILL,000)                         TO                +
         KEY=YES                                                               +
         SIGN=NO                                                               +
         SNAME='FILLER'                                                        +
         RELATED=NO                                                            +
         COLUMN=11                                                             +
         AUDIT=NO                                                              +
         CAUDIT=NO                                                             +
         MSG=NO                                                                ;
AR       ADFSEG01 (T, ,SAMPIV,000)                           ADFSEG_HAS        +
         ADFDTE01 (T, ,SAIVFILL,000)                                           ;
AR       ADFDTE01 (T, ,SAIVFILL,000)                         POINT_DTE         +
         ELEMENT  (T,C,DIV-FILL,000)                                           ;
*
```

```
*         ADFDTE01:  SAIVPRIC
*         ELEMENT :  UNIT-PRICE
*
A         ADFDTE01 (T, ,SAIVPRIC,000)                                            ;
CI        ADFDTE01 (T, ,SAIVPRIC,000)                           TO               +
          KEY=NO                                                                 +
          SIGN=YES                                                               +
          SNAME='UNIT PRICE'                                                     +
          RELATED=NO                                                             +
          AUDIT=NO                                                               +
          CAUDIT=NO                                                              +
          MSG=NO                                                                 ;
AR        ADFSEG01 (T, ,SAMPIV,000)                             ADFSEG_HAS       +
          ADFDTE01 (T, ,SAIVPRIC,000)                                            ;
AR        ADFDTE01 (T, ,SAIVPRIC,000)                           POINT_DTE        +
          ELEMENT  (T,C,UNIT-PRICE,000)                                          ;
*
*         ADFDTE01:  SAIVRPRI
*         ELEMENT :  R-PRICE
*
A         ADFDTE01 (T, ,SAIVRPRI,000)                                            ;
CI        ADFDTE01 (T, ,SAIVRPRI,000)                           TO               +
          KEY=NO                                                                 +
          SIGN=YES                                                               +
          SNAME=' '                                                              +
          RELATED=YES                                                            +
          COLUMN=19                                                              +
          AUDIT=NO                                                               +
          CAUDIT=NO                                                              +
          MSG=NO                                                                 ;
AR        ADFSEG01 (T, ,SAMPIV,000)                             ADFSEG_HAS       +
          ADFDTE01 (T, ,SAIVRPRI,000)                                            ;
AR        ADFDTE01 (T, ,SAIVRPRI,000)                           POINT_DTE        +
          ELEMENT  (T,C,R-PRICE,000)                                             ;
*
*         ADFDTE01:  SAIVUNIT
*         ELEMENT :  UNIT
*
A         ADFDTE01 (T, ,SAIVUNIT,000)                                            ;
CI        ADFDTE01 (T, ,SAIVUNIT,000)                           TO               +
          KEY=NO                                                                 +
          SIGN=NO                                                                +
          SNAME='UNIT'                                                           +
          RELATED=NO                                                             +
          AUDIT=NO                                                               +
          CAUDIT=NO                                                              +
          MSG=NO                                                                 ;
AR        ADFSEG01 (T, ,SAMPIV,000)                             ADFSEG_HAS       +
          ADFDTE01 (T, ,SAIVUNIT,000)                                            ;
AR        ADFDTE01 (T, ,SAIVUNIT,000)                           POINT_DTE        +
          ELEMENT  (T,C,UNIT,000)                                                ;
*
*         ADFDTE01:  SAIVCOAP
*         ELEMENT :  ATTR-COAP
*
A         ADFDTE01 (T, ,SAIVCOAP,000)                                            ;
CI        ADFDTE01 (T, ,SAIVCOAP,000)                           TO               +
          KEY=NO                                                                 +
          SIGN=YES                                                               +
          SNAME='ATTR COAP'                                                      +
          RELATED=NO                                                             +
          AUDIT=NO                                                               +
          CAUDIT=NO                                                              +
          MSG=NO                                                                 ;
AR        ADFSEG01 (T, ,SAMPIV,000)                             ADFSEG_HAS       +
          ADFDTE01 (T, ,SAIVCOAP,000)                                            ;
AR        ADFDTE01 (T, ,SAIVCOAP,000)                           POINT_DTE        +
          ELEMENT  (T,C,ATTR-COAP,000)                                           ;
*
*         ADFDTE01:  SAIVPLAN
*         ELEMENT :  ATTR-PLANNED
*
A         ADFDTE01 (T, ,SAIVPLAN,000)                                            ;
CI        ADFDTE01 (T, ,SAIVPLAN,000)                           TO               +
          KEY=NO                                                          +
          SIGN=YES                                                        +
          SNAME='ATTR PLANNED'                                            +
          RELATED=NO                                                      +
          AUDIT=NO                                                        +
          CAUDIT=NO                                                       +
          MSG=NO                                                          ;
AR        ADFSEG01 (T, ,SAMPIV,000)                     ADFSEG_HAS       ,+
          ADFDTE01 (T, ,SAIVPLAN,000)                                     ;
AR        ADFDTE01 (T, ,SAIVPLAN,000)                   POINT_DTE         +
          ELEMENT  (T,C,ATTR-PLANNED,000)                                 ;
*
*         ADFDTE01:  SAIVCOAD
*         ELEMENT :  ATTR-COAD
*
A         ADFDTE01 (T, ,SAIVCOAD,000)                                     ;
CI        ADFDTE01 (T, ,SAIVCOAD,000)                   TO                +
          KEY=NO                                                          +
          SIGN=NO                                                         +
          SNAME='ATTR COAD'                                               +
          RELATED=NO                                                      +
          AUDIT=NO                                                        +
          CAUDIT=NO                                                       +
          MSG=NO                                                          ;
AR        ADFSEG01 (T, ,SAMPIV,000)                     ADFSEG_HAS        +
          ADFDTE01 (T, ,SAIVCOAD,000)                                     ;
AR        ADFDTE01 (T, ,SAIVCOAD,000)                   POINT_DTE         +
          ELEMENT  (T,C,ATTR-COAD,000)                                    ;
*
*         ADFDTE01:  SAIVCDAY
*         ELEMENT :  STOCK-DATE
*
A         ADFDTE01 (T, ,SAIVCDAY,000)                                     ;
CI        ADFDTE01 (T, ,SAIVCDAY,000)                   TO                +
          KEY=NO                                                          +
          SIGN=NO                                                         +
          SNAME='STOCK DATE'                                              +
          RELATED=NO                                                      +
          AUDIT=NO                                                        +
          CAUDIT=NO                                                       +
          MSG=NO                                                          ;
AR        ADFSEG01 (T, ,SAMPIV,000)                     ADFSEG_HAS        +
          ADFDTE01 (T, ,SAIVCDAY,000)                                     ;
AR        ADFDTE01 (T, ,SAIVCDAY,000)                   POINT_DTE         +
          ELEMENT  (T,C,STOCK-DATE,000)                                   ;
*
*         ADFDTE01:  SAIVTDAY
*         ELEMENT :  LAST-TRANS
*
A         ADFDTE01 (T, ,SAIVTDAY,000)                                     ;
CI        ADFDTE01 (T, ,SAIVTDAY,000)                   TO                +
          KEY=NO                                                          +
          SIGN=YES                                                        +
          SNAME='LAST TRANS'                                              +
          RELATED=NO                                                      +
          AUDIT=NO                                                        +
          CAUDIT=NO                                                       +
          MSG=NO                                                          ;
AR        ADFSEG01 (T, ,SAMPIV,000)                     ADFSEG_HAS        +
          ADFDTE01 (T, ,SAIVTDAY,000)                                     ;
AR        ADFDTE01 (T, ,SAIVTDAY,000)                   POINT_DTE         +
          ELEMENT  (T,C,LAST-TRANS,000)                                   ;
*
*         ADFDTE01:  SAIVREQC
*         ELEMENT :  RQMNTS-CURRENT
*
A         ADFDTE01 (T, ,SAIVREQC,000)                                     ;
CI        ADFDTE01 (T, ,SAIVREQC,000)                   TO                +
          KEY=NO                                                          +
          SIGN=YES                                                        +
          SNAME='RQMNTS CURRENT'                                          +
          RELATED=NO                                                      +
          AUDIT=NO                                                        +
          CAUDIT=NO                                                       +
         MSG=NO                                                     ;
AR       ADFSEG01 (T, ,SAMPIV,000)                     ADFSEG_HAS   +
         ADFDTE01 (T, ,SAIVREQC,000)                                ;
AR       ADFDTE01 (T, ,SAIVREQC,000)                   POINT_DTE    +
         ELEMENT  (T,C,RQMNTS-CURRENT,000)                          ;
*
*        ADFDTE01:  SAIVRREQ
*        ELEMENT :  R-RQMNTS
*
A        ADFDTE01 (T, ,SAIVRREQ,000)                                ;
CI       ADFDTE01 (T, ,SAIVRREQ,000)                   TO           +
         KEY=NO                                                     +
         SIGN=YES                                                   +
         SNAME=' '                                                  +
         RELATED=YES                                                +
         COLUMN=29                                                  +
         AUDIT=NO                                                   +
         CAUDIT=NO                                                  +
         MSG=NO                                                     ;
AR       ADFSEG01 (T, ,SAMPIV,000)                     ADFSEG_HAS   +
         ADFDTE01 (T, ,SAIVRREQ,000)                                ;
AR       ADFDTE01 (T, ,SAIVRREQ,000)                   POINT_DTE    +
         ELEMENT  (T,C,R-RQMNTS,000)                                ;
*
*        ADFDTE01:  SAIVREQU
*        ELEMENT :  RQMNTS-UNPLAN
*
A        ADFDTE01 (T, ,SAIVREQU,000)                                ;
CI       ADFDTE01 (T, ,SAIVREQU,000)                   TO           +
         KEY=NO                                                     +
         SIGN=YES                                                   +
         SNAME='RQMNTS UNPLAN'                                      +
         RELATED=NO                                                 +
         AUDIT=NO                                                   +
         CAUDIT=NO                                                  +
         MSG=NO                                                     ;
AR       ADFSEG01 (T, ,SAMPIV,000)                     ADFSEG_HAS   +
         ADFDTE01 (T, ,SAIVREQU,000)                                ;
AR       ADFDTE01 (T, ,SAIVREQU,000)                   POINT_DTE    +
         ELEMENT  (T,C,RQMNTS-UNPLAN,000)                           ;
*
*        ADFDTE01:  SAIVONOR
*        ELEMENT :  ON-ORDER
*
A        ADFDTE01 (T, ,SAIVONOR,000)                                ;
CI       ADFDTE01 (T, ,SAIVONOR,000)                   TO           +
         KEY=NO                                                     +
         SIGN=YES                                                   +
         SNAME='ON ORDER'                                           +
         RELATED=NO                                                 +
         AUDIT=NO                                                   +
         CAUDIT=NO                                                  +
         MSG=NO                                                     ;
AR       ADFSEG01 (T, ,SAMPIV,000)                     ADFSEG_HAS   +
         ADFDTE01 (T, ,SAIVONOR,000)                                ;
AR       ADFDTE01 (T, ,SAIVONOR,000)                   POINT_DTE    +
         ELEMENT  (T,C,ON-ORDER,000)                                ;
*
*        ADFDTE01:  SAIVRONO
*        ELEMENT :  R-ON-ORDER
*
A        ADFDTE01 (T, ,SAIVRONO,000)                                ;
CI       ADFDTE01 (T, ,SAIVRONO,000)                   TO           +
         KEY=NO                                                     +
         SIGN=YES                                                   +
         SNAME=' '                                                  +
         RELATED=YES                                                +
         COLUMN=38                                                  +
         AUDIT=NO                                                   +
         CAUDIT=NO                                                  +
         MSG=NO                                                     ;
AR       ADFSEG01 (T, ,SAMPIV,000)                     ADFSEG_HAS   +
         ADFDTE01 (T, ,SAIVRONO,000)                                ;
AR       ADFDTE01 (T, ,SAIVRONO,000)                   POINT_DTE    +
          ELEMENT  (T,C,R-ON-ORDER,000)                                    ;
*
*         ADFDTE01:  SAIVSTCK
*         ELEMENT :  TOTAL-STOCK
*
A         ADFDTE01 (T, ,SAIVSTCK,000)                                      ;
CI        ADFDTE01 (T, ,SAIVSTCK,000)                  TO                  +
          KEY=NO                                                           +
          SIGN=YES                                                         +
          SNAME='TOTAL STOCK'                                              +
          RELATED=NO                                                       +
          AUDIT=NO                                                         +
          CAUDIT=NO                                                        +
          MSG=NO                                                           ;
AR        ADFSEG01 (T, ,SAMPIV,000)                    ADFSEG_HAS          +
          ADFDTE01 (T, ,SAIVSTCK,000)                                      ;
AR        ADFDTE01 (T, ,SAIVSTCK,000)                  POINT_DTE           +
          ELEMENT  (T,C,TOTAL-STOCK,000)                                   ;
*
*         ADFDTE01:  SAIVRSTC
*         ELEMENT :  R-TOTAL-STOCK
*
A         ADFDTE01 (T, ,SAIVRSTC,000)                                      ;
CI        ADFDTE01 (T, ,SAIVRSTC,000)                  TO                  +
          KEY=NO                                                           +
          SIGN=YES                                                         +
          SNAME=' '                                                        +
          RELATED=YES                                                      +
          COLUMN=47                                                        +
          AUDIT=NO                                                         +
          CAUDIT=NO                                                        +
          MSG=NO                                                           ;
AR        ADFSEG01 (T, ,SAMPIV,000)                    ADFSEG_HAS          +
          ADFDTE01 (T, ,SAIVRSTC,000)                                      ;
AR        ADFDTE01 (T, ,SAIVRSTC,000)                     POINT_DTE        +
          ELEMENT  (T,C,R-TOTAL-STOCK,000)                                 ;
*
*         ADFDTE01:  SAIVDIPL
*         ELEMENT :  DISB-PLAN
*
A         ADFDTE01 (T, ,SAIVDIPL,000)                                      ;
CI        ADFDTE01 (T, ,SAIVDIPL,000)                  TO                  +
          KEY=NO                                                           +
          SIGN=YES                                                         +
          SNAME='DISB PLAN'                                                +
          RELATED=NO                                                       +
          AUDIT=NO                                                         +
          CAUDIT=NO                                                        +
          MSG=NO                                                           ;
AR        ADFSEG01 (T, ,SAMPIV,000)                    ADFSEG_HAS          +
          ADFDTE01 (T, ,SAIVDIPL,000)                                      ;
AR        ADFDTE01 (T, ,SAIVDIPL,000)                  POINT_DTE           +
          ELEMENT  (T,C,DISB-PLAN,000)                                     ;
*
*         ADFDTE01:  SAIVRDIP
*         ELEMENT :  R-DISB-PLAN
*
A         ADFDTE01 (T, ,SAIVRDIP,000)                                      ;
CI        ADFDTE01 (T, ,SAIVRDIP,000)                  TO                  +
          KEY=NO                                                           +
          SIGN=YES                                                         +
          SNAME=' '                                                        +
          RELATED=YES                                                      +
          COLUMN=56                                                        +
          AUDIT=NO                                                         +
          CAUDIT=NO                                                        +
          MSG=NO                                                           ;
AR        ADFSEG01 (T, ,SAMPIV,000)                    ADFSEG_HAS          +
          ADFDTE01 (T, ,SAIVRDIP,000)                                      ;
AR        ADFDTE01 (T, ,SAIVRDIP,000)                  POINT_DTE           +
          ELEMENT  (T,C,R-DISB-PLAN,000)                                   ;
*
*         ADFDTE01:  SAIVDIUN
*         ELEMENT :  DISB-UNPLAN
```

```
*
A       ADFDTE01 (T, ,SAIVDIUN,000)                               ;
CI      ADFDTE01 (T, ,SAIVDIUN,000)               TO              +
        KEY=NO                                                    +
        SIGN=YES                                                  +
        SNAME='DISB UNPLAN'                                       +
        RELATED=NO                                                +
        AUDIT=NO                                                  +
        CAUDIT=NO                                                 +
        MSG=NO                                                    ;
AR      ADFSEG01 (T, ,SAMPIV,000)                 ADFSEG_HAS      +
        ADFDTE01 (T, ,SAIVDIUN,000)                               ;
AR      ADFDTE01 (T, ,SAIVDIUN,000)               POINT_DTE       +
        ELEMENT  (T,C,DISB-UNPLAN,000)                            ;
*
*       ADFDTE01:  SAIVRDIU
*       ELEMENT :  R-DISB-UNPLAN
*
A       ADFDTE01 (T, ,SAIVRDIU,000)                               ;
CI      ADFDTE01 (T, ,SAIVRDIU,000)               TO              +
        KEY=NO                                                    +
        SIGN=YES                                                  +
        SNAME=' '                                                 +
        RELATED=YES                                               +
        COLUMN=65                                                 +
        AUDIT=NO                                                  +
        CAUDIT=NO                                                 +
        MSG=NO                                                    ;
AR      ADFSEG01 (T, ,SAMPIV,000)                 ADFSEG_HAS      +
        ADFDTE01 (T, ,SAIVRDIU,000)                               ;
AR      ADFDTE01 (T, ,SAIVRDIU,000)               POINT_DTE       +
        ELEMENT  (T,C,R-DISB-UNPLAN,000)                          ;
*
*       ADFDTE01:  SAIVDISP
*       ELEMENT :  DISB-SPARES
*
A       ADFDTE01 (T, ,SAIVDISP,000)                               ;
CI      ADFDTE01 (T, ,SAIVDISP,000)               TO              +
        KEY=NO                                                    +
        SIGN=YES                                                  +
        SNAME='DISB SPARES'                                       +
        RELATED=NO                                                +
        AUDIT=NO                                                  +
        CAUDIT=NO                                                 +
        MSG=NO                                                    ;
AR      ADFSEG01 (T, ,SAMPIV,000)                 ADFSEG_HAS      +
        ADFDTE01 (T, ,SAIVDISP,000)                               ;
AR      ADFDTE01 (T, ,SAIVDISP,000)               POINT_DTE       +
        ELEMENT  (T,C,DISB-SPARES,000)                            ;
*
*       ADFDTE01:  SAIVDIDV
*       ELEMENT :  DISB-DIVERS
*
A       ADFDTE01 (T, ,SAIVDIDV,000)                               ;
CI      ADFDTE01 (T, ,SAIVDIDV,000)               TO              +
        KEY=NO                                                    +
        SIGN=YES                                                  +
        SNAME='DISB DIVERS'                                       +
        RELATED=NO                                                +
        AUDIT=NO                                                  +
        CAUDIT=NO                                                 +
        MSG=NO                                                    ;
AR      ADFSEG01 (T, ,SAMPIV,000)                 ADFSEG_HAS      +
        ADFDTE01 (T, ,SAIVDIDV,000)                               ;
AR      ADFDTE01 (T, ,SAIVDIDV,000)               POINT_DTE       +
        ELEMENT  (T,C,DISB-DIVERS,000)                            ;
*******************************************************************************
*
*       ADFSEG01:  CY
*       SEGMENT :  CYCCOUNT
*
*******************************************************************************
A       ADFSEG01 (T, ,SAMPCY,000)                                 ;
CI      ADFSEG01 (T, ,SAMPCY,000)                 TO              +
         PCBNO=001                                                       +
         DBID=PA                                                         +
         TRAILER=01                                                      +
         SKSEGS=37                                                       +
         KASCEND=YES                                                     +
         ADBSNAME=DI21PART                                               +
         ALENGTH=00025                                                   ;
AR       ADFSYS01 (T, ,SAMP,000)                          ADFSYS_HAS     +
         ADFSEG01 (T, ,SAMPCY,000)                                       ;
AR       ADFSEG01 (T, ,SAMPCY,000)                        POINT_SEG      +
         SEGMENT  (T,A,CYCCOUNT,000)                                     ;
*
*        ADFDTE01:  SACYKEY
*        ELEMENT :  CYCLKEY
*
A        ADFDTE01 (T, ,SACYKEY,000)                                      ;
CI       ADFDTE01 (T, ,SACYKEY,000)                       TO             +
         KEY=YES                                                         +
         SIGN=NO                                                         +
         SNAME='20'                                                      +
         RELATED=NO                                                      +
         AUDIT=NO                                                        +
         CAUDIT=NO                                                       +
         MSG=NO                                                          ;
AR       ADFSEG01 (T, ,SAMPCY,000)                        ADFSEG_HAS     +
         ADFDTE01 (T, ,SACYKEY,000)                                      ;
AR       ADFDTE01 (T, ,SACYKEY,000)                       POINT_DTE      +
         ELEMENT  (T,C,CYCLKEY,000)                                      ;
*
*        ADFDTE01:  SACYCNTA
*        ELEMENT :  PHYS-COUNT
*
A        ADFDTE01 (T, ,SACYCNTA,000)                                     ;
CI       ADFDTE01 (T, ,SACYCNTA,000)                      TO             +
         KEY=NO                                                          +
         SIGN=YES                                                        +
         SNAME='PHYS COUNT'                                              +
         RELATED=NO                                                      +
         AUDIT=NO                                                        +
         CAUDIT=NO                                                       +
         MSG=NO                                                          ;
AR       ADFSEG01 (T, ,SAMPCY,000)                        ADFSEG_HAS     +
         ADFDTE01 (T, ,SACYCNTA,000)                                     ;
AR       ADFDTE01 (T, ,SACYCNTA,000)                      POINT_DTE      +
         ELEMENT  (T,C,PHYS-COUNT,000)                                   ;
*
*        ADFDTE01:  SACYSTCK
*        ELEMENT :  BOOK-COUNT
*
A        ADFDTE01 (T, ,SACYSTCK,000)                                     ;
CI       ADFDTE01 (T, ,SACYSTCK,000)                      TO             +
         KEY=NO                                                          +
         SIGN=YES                                                        +
         SNAME='BOOK-COUNT'                                              +
         RELATED=NO                                                      +
         AUDIT=NO                                                        +
         CAUDIT=NO                                                       +
         MSG=NO                                                          ;
AR       ADFSEG01 (T, ,SAMPCY,000)                        ADFSEG_HAS     +
         ADFDTE01 (T, ,SACYSTCK,000)                                     ;
AR       ADFDTE01 (T, ,SACYSTCK,000)                      POINT_DTE      +
         ELEMENT  (T,C,BOOK-COUNT,000)                                   ;
```

APPENDIX B. ADFOUT PROCESSOR SAMPLE PROCEDURE OUTPUT

```
**********************************************************************
*
*  SYSTEM:  SAMP          DATE:   08/14/84    TIME:   13:00:28
*
**********************************************************************
   SYSTEM   SYSID=SAMP,
            ADFID=MFC1,
            SOMTX=OR,
            PGROUP=ZZ,
            PCBNO=001,
            DBID=PA,
            STRAILER=1,
            SHEADING='S A M P L E   P R O B L E M',
            SFORMAT=DASH
**********************************************************************
*
*  SEGMENT: SEGID= PA     DATE:   08/14/84    TIME:   13:00:28
*           DICTIONARY ADF SEG: T SAMPPA                           000
*           DICTIONARY SEGMENT: TCPARTROOT                         000
*
**********************************************************************
   SEGMENT  ID=PA,
            PARENT=0,
            NAME=PARTROOT,
            LENGTH=00050,
            KEYNAME=PARTKEY,
            PCBNO=001,
            DBID=PA,
            TRAILER=01,
            SKSEGS=18,
            KASCEND=YES
*
*           DICTIONARY ADF FLD: T SAPAKEY                          000
*           DICTIONARY   FIELD: TCPART-NUMBER                      000
*
   FIELD    ID=KEY,
            TYPE=ALPHANUM,
            LENGTH=00017,
            POSITION=00001,
            KEY=YES,
            SNAME='PART NUMBER',
            RELATE=NO,
            SIGN=NO,
            AUDIT=NO,
            CAUDIT=NO,
            MSG=NO
*
*           DICTIONARY ADF FLD: T SAPADESC                         000
*           DICTIONARY   FIELD: TCPART-DESC                        000
*
   FIELD    ID=DESC,
            TYPE=ALPHANUM,
            LENGTH=00020,
            POSITION=00027,
            KEY=NO,
            SNAME='DESCRIPTION',
            RELATE=YES,
            SIGN=NO,
            AUDIT=NO,
            CAUDIT=NO,
            MSG=NO
```

```
*********************************************************************
*
*   SEGMENT: SEGID= PD      DATE:    08/14/84     TIME:    13:00:28
*            DICTIONARY ADF SEG: T SAMPPD                          000
*            DICTIONARY SEGMENT: TCSTANINFO                        000
*
*********************************************************************
   SEGMENT   ID=PD,
             PARENT=PA,
             NAME=STANINFO,
             LENGTH=00085,
             KEYNAME=STANKEY,
             PCBNO=001,
             DBID=PA,
             TRAILER=01,
             SKSEGS=37,
             KASCEND=YES
*
*            DICTIONARY ADF FLD: T SAPDKEY                         000
*            DICTIONARY   FIELD: TCSTANKEY                         000
*
   FIELD     ID=KEY,
             TYPE=ALPHANUM,
             LENGTH=00002,
             POSITION=00001,
             KEY=YES,
             SNAME='KEY FIELD',
             RELATE=NO,
             SIGN=NO,
             AUDIT=NO,
             CAUDIT=NO,
             MSG=NO
*
*            DICTIONARY ADF FLD: T SAPDPRCD                        000
*            DICTIONARY   FIELD: TCPROC-CODE                       000
*
   FIELD     ID=PRCD,
             TYPE=ALPHANUM,
             LENGTH=00002,
             POSITION=00019,
             KEY=NO,
             SNAME='PROC CODE',
             RELATE=NO,
             SIGN=NO,
             AUDIT=NO,
             CAUDIT=NO,
             MSG=NO
*
*            DICTIONARY ADF FLD: T SAPDINVC                        000
*            DICTIONARY   FIELD: TCINV-CODE                        000
*
   FIELD     ID=INVC,
             TYPE=ALPHANUM,
             LENGTH=00001,
             POSITION=00021,
             KEY=NO,
             SNAME='INVENTORY CODE',
             RELATE=NO,
             SIGN=NO,
             AUDIT=NO,
             CAUDIT=NO,
             MSG=NO
```

```
*
*            DICTIONARY ADF FLD: T SAPDPLRV                                    000
*            DICTIONARY   FIELD: TCPLAN-REV-NO                                 000
*
   FIELD     ID=PLRV,
             TYPE=ALPHANUM,
             LENGTH=00002,
             POSITION=00022,
             KEY=NO,
             SNAME='PLAN REV NO',
             RELATE=NO,
             SIGN=NO,
             AUDIT=NO,
             CAUDIT=NO,
             MSG=NO
*
*            DICTIONARY ADF FLD: T SAPDMKDP                                    000
*            DICTIONARY   FIELD: TCMAKE-DEPT                                   000
*
   FIELD     ID=MKDP,
             TYPE=ALPHANUM,
             LENGTH=00004,
             POSITION=00048,
             KEY=NO,
             SNAME='MAKE DEPT',
             RELATE=NO,
             SIGN=NO,
             AUDIT=NO,
             CAUDIT=NO,
             MSG=NO
*
*            DICTIONARY ADF FLD: T SAPDCOMM                                    000
*            DICTIONARY   FIELD: TCCOMM-CODE                                   000
*
   FIELD     ID=COMM,
             TYPE=ALPHANUM,
             LENGTH=00004,
             POSITION=00054,
             KEY=NO,
             SNAME='COMM CODE',
             RELATE=NO,
             SIGN=NO,
             AUDIT=NO,
             CAUDIT=NO,
             MSG=NO
*
*            DICTIONARY ADF FLD: T SAPDRISP                                    000
*            DICTIONARY   FIELD: TCRIGHT-MAKE-TIME                             000
*
   FIELD     ID=RISP,
             TYPE=DEC,
             LENGTH=00002,
             POSITION=00062,
             KEY=NO,
             SNAME='RIGHT MAKE TIME',
             RELATE=NO,
             SIGN=YES,
             AUDIT=NO,
             CAUDIT=NO,
             MSG=NO
```

```
*
*          DICTIONARY ADF FLD: T SAPDWRSP                              000
*          DICTIONARY   FIELD: TCWRONG-MAKE-TIME                       000
*
   FIELD   ID=WRSP,
           TYPE=DEC,
           LENGTH=00002,
           POSITION=00071,
           KEY=NO,
           SNAME='WRONG MAKE TIME',
           RELATE=NO,
           SIGN=YES,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
**********************************************************************
*
*  SEGMENT: SEGID= IV     DATE:   08/14/84    TIME:   13:00:28
*          DICTIONARY ADF SEG: T SAMPIV                                000
*          DICTIONARY SEGMENT: TCSTOKSTAT                              000
*
**********************************************************************
   SEGMENT ID=IV,
           PARENT=PA,
           NAME=STOKSTAT,
           LENGTH=00160,
           KEYNAME=STOCKEY,
           PCBNO=001,
           DBID=PA,
           TRAILER=01,
           SKLEFT='INVENTORY        UNIT      CURRENT',
           SKLEFT='LOCATION         PRICE     REQMNTS',
           SKRIGHT='   ON      TOTAL      DISBURSEMENTS',
           SKRIGHT=' ORDER    STOCK   PLANNED  UNPLANNED',
           SKSEGS=37,
           KASCEND=YES
*
*          DICTIONARY ADF FLD: T SAIVW                                 000
*          DICTIONARY   FIELD: TCFILL-0                                000
*
   FIELD   ID=W,
           TYPE=ALPHANUM,
           LENGTH=00002,
           POSITION=00001,
           KEY=YES,
           SNAME='00',
           RELATE=NO,
           COLUMN=01,
           SIGN=NO,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
*
*          DICTIONARY ADF FLD: T SAIVAREA                              000
*          DICTIONARY   FIELD: TCAREA                                  000
*
```

```
    FIELD    ID=AREA,
             TYPE=ALPHANUM,
             LENGTH=00001,
             POSITION=00003,
             KEY=YES,
             SNAME='AREA',
             RELATE=NO,
             COLUMN=03,
             SIGN=NO,
             AUDIT=NO,
             CAUDIT=NO,
             MSG=NO
*
*            DICTIONARY ADF FLD: T SAIVINVD                        000
*            DICTIONARY   FIELD: TCINV-DEPT                        000
*
    FIELD    ID=INVD,
             TYPE=ALPHANUM,
             LENGTH=00002,
             POSITION=00004,
             KEY=YES,
             SNAME='INV DEPT',
             RELATE=NO,
             COLUMN=04,
             SIGN=NO,
             AUDIT=NO,
             CAUDIT=NO,
             MSG=NO
*
*            DICTIONARY ADF FLD: T SAIVPROJ                        000
*            DICTIONARY   FIELD: TCPROJECT                         000
*
    FIELD    ID=PROJ,
             TYPE=ALPHANUM,
             LENGTH=00003,
             POSITION=00006,
             KEY=YES,
             SNAME='PROJECT',
             RELATE=NO,
             COLUMN=06,
             SIGN=NO,
             AUDIT=NO,
             CAUDIT=NO,
             MSG=NO
*
*            DICTIONARY ADF FLD: T SAIVDIV                         000
*            DICTIONARY   FIELD: TCDIVISION                        000
*
    FIELD    ID=DIV,
             TYPE=ALPHANUM,
             LENGTH=00002,
             POSITION=00009,
             KEY=YES,
             SNAME='DIVISION',
             RELATE=NO,
             COLUMN=10,
             SIGN=NO,
             AUDIT=NO,
             CAUDIT=NO,
             MSG=NO
```

```
*
*          DICTIONARY ADF FLD: T SAIVFILL                        000
*          DICTIONARY   FIELD: TCDIV-FILL                        000
*
   FIELD   ID=FILL,
           TYPE=ALPHANUM,
           LENGTH=00006,
           POSITION=00011,
           KEY=YES,
           SNAME='FILLER',
           RELATE=NO,
           COLUMN=11,
           SIGN=NO,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
*
*          DICTIONARY ADF FLD: T SAIVPRIC                        000
*          DICTIONARY   FIELD: TCUNIT-PRICE                      000
*
   FIELD   ID=PRIC,
           TYPE=DEC,
           LENGTH=00009,
           POSITION=00021,
           DECIMAL=0002,
           KEY=NO,
           SNAME='UNIT PRICE',
           RELATE=NO,
           SIGN=YES,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
*
*          DICTIONARY ADF FLD: T SAIVRPRI                        000
*          DICTIONARY   FIELD: TCR-PRICE                         000
*
   FIELD   ID=RPRI,
           TYPE=DEC,
           LENGTH=00007,
           POSITION=00023,
           DECIMAL=0002,
           KEY=NO,
           SNAME=' ',
           RELATE=YES,
           COLUMN=19,
           SIGN=YES,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
*
*          DICTIONARY ADF FLD: T SAIVUNIT                        000
*          DICTIONARY   FIELD: TCUNIT                            000
*
   FIELD   ID=UNIT,
           TYPE=ALPHANUM,
           LENGTH=00004,
           POSITION=00035,
           KEY=NO,
           SNAME='UNIT',
           RELATE=NO,
           SIGN=NO,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
```

```
*
*          DICTIONARY ADF FLD: T SAIVCOAP                                    000
*          DICTIONARY   FIELD: TCATTR-COAP                                   000
*
   FIELD   ID=COAP,
           TYPE=DEC,
           LENGTH=00003,
           POSITION=00051,
           KEY=NO,
           SNAME='ATTR COAP',
           RELATE=NO,
           SIGN=YES,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
*
*          DICTIONARY ADF FLD: T SAIVPLAN                                    000
*          DICTIONARY   FIELD: TCATTR-PLANNED                                000
*
   FIELD   ID=PLAN,
           TYPE=DEC,
           LENGTH=00003,
           POSITION=00054,
           KEY=NO,
           SNAME='ATTR PLANNED',
           RELATE=NO,
           SIGN=YES,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
*
*          DICTIONARY ADF FLD: T SAIVCOAD                                    000
*          DICTIONARY   FIELD: TCATTR-COAD                                   000
*
   FIELD   ID=COAD,
           TYPE=ALPHANUM,
           LENGTH=00001,
           POSITION=00057,
           KEY=NO,
           SNAME='ATTR COAD',
           RELATE=NO,
           SIGN=NO,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
*
*          DICTIONARY ADF FLD: T SAIVCDAY                                    000
*          DICTIONARY   FIELD: TCSTOCK-DATE                                  000
*
   FIELD   ID=CDAY,
           TYPE=ALPHANUM,
           LENGTH=00003,
           POSITION=00072,
           KEY=NO,
           SNAME='STOCK DATE',
           RELATE=NO,
           SIGN=NO,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
```

```
*
*          DICTIONARY ADF FLD: T SAIVTDAY                          000
*          DICTIONARY   FIELD: TCLAST-TRANS                        000
*
   FIELD   ID=TDAY,
           TYPE=DEC,
           LENGTH=00003,
           POSITION=00075,
           KEY=NO,
           SNAME='LAST TRANS',
           RELATE=NO,
           SIGN=YES,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
*
*          DICTIONARY ADF FLD: T SAIVREQC                          000
*          DICTIONARY   FIELD: TCRQMNTS-CURRENT                    000
*
   FIELD   ID=REQC,
           TYPE=DEC,
           LENGTH=00007,
           POSITION=00090,
           KEY=NO,
           SNAME='RQMNTS CURRENT',
           RELATE=NO,
           SIGN=YES,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
*
*          DICTIONARY ADF FLD: T SAIVRREQ                          000
*          DICTIONARY   FIELD: TCR-RQMNTS                          000
*
   FIELD   ID=RREQ,
           TYPE=DEC,
           LENGTH=00005,
           POSITION=00092,
           KEY=NO,
           SNAME=' ',
           RELATE=YES,
           COLUMN=29,
           SIGN=YES,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
```

```
*
*          DICTIONARY ADF FLD: T SAIVREQU                                000
*          DICTIONARY   FIELD: TCRQMNTS-UNPLAN                           000
*
   FIELD   ID=REQU,
           TYPE=DEC,
           LENGTH=00007,
           POSITION=00098,
           KEY=NO,
           SNAME='RQMNTS UNPLAN',
           RELATE=NO,
           SIGN=YES,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
*
*          DICTIONARY ADF FLD: T SAIVONOR                                000
*          DICTIONARY   FIELD: TCON-ORDER                                000
*
   FIELD   ID=ONOR,
           TYPE=DEC,
           LENGTH=00007,
           POSITION=00106,
           KEY=NO,
           SNAME='ON ORDER',
           RELATE=NO,
           SIGN=YES,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
*
*          DICTIONARY ADF FLD: T SAIVRONO                                000
*          DICTIONARY   FIELD: TCR-ON-ORDER                              000
*
   FIELD   ID=RONO,
           TYPE=DEC,
           LENGTH=00005,
           POSITION=00108,
           KEY=NO,
           SNAME=' ',
           RELATE=YES,
           COLUMN=38,
           SIGN=YES,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
```

```
*
*             DICTIONARY ADF FLD: T SAIVSTCK                              000
*             DICTIONARY   FIELD: TCTOTAL-STOCK                           000
*
   FIELD      ID=STCK,
              TYPE=DEC,
              LENGTH=00007,
              POSITION=00114,
              KEY=NO,
              SNAME='TOTAL STOCK',
              RELATE=NO,
              SIGN=YES,
              AUDIT=NO,
              CAUDIT=NO,
              MSG=NO
*
*             DICTIONARY ADF FLD: T SAIVRSTC                              000
*             DICTIONARY   FIELD: TCR-TOTAL-STOCK                         000
*
   FIELD      ID=RSTC,
              TYPE=DEC,
              LENGTH=00005,
              POSITION=00116,
              KEY=NO,
              SNAME=' ',
              RELATE=YES,
              COLUMN=47,
              SIGN=YES,
              AUDIT=NO,
              CAUDIT=NO,
              MSG=NO
*
*             DICTIONARY ADF FLD: T SAIVDIPL                              000
*             DICTIONARY   FIELD: TCDISB-PLAN                             000
*
   FIELD      ID=DIPL,
              TYPE=DEC,
              LENGTH=00007,
              POSITION=00122,
              KEY=NO,
              SNAME='DISB PLAN',
              RELATE=NO,
              SIGN=YES,
              AUDIT=NO,
              CAUDIT=NO,
              MSG=NO
*
*             DICTIONARY ADF FLD: T SAIVRDIP                              000
*             DICTIONARY   FIELD: TCR-DISB-PLAN                           000
*
   FIELD      ID=RDIP,
              TYPE=DEC,
              LENGTH=00005,
              POSITION=00124,
              KEY=NO,
              SNAME=' ',
              RELATE=YES,
              COLUMN=56,
              SIGN=YES,
              AUDIT=NO,
              CAUDIT=NO,
              MSG=NO
```

```
*                                                                          000
*          DICTIONARY ADF FLD: T SAIVDIUN                                  000
*          DICTIONARY   FIELD: TCDISB-UNPLAN
*
   FIELD   ID=DIUN,
           TYPE=DEC,
           LENGTH=00007,
           POSITION=00130,
           KEY=NO,
           SNAME='DISB UNPLAN',
           RELATE=NO,
           SIGN=YES,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
*                                                                          000
*          DICTIONARY ADF FLD: T SAIVRDIU                                  000
*          DICTIONARY   FIELD: TCR-DISB-UNPLAN
*
   FIELD   ID=RDIU,
           TYPE=DEC,
           LENGTH=00005,
           POSITION=00132,
           KEY=NO,
           SNAME=' ',
           RELATE=YES,
           COLUMN=65,
           SIGN=YES,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
*                                                                          000
*          DICTIONARY ADF FLD: T SAIVDISP                                  000
*          DICTIONARY   FIELD: TCDISB-SPARES
*
   FIELD   ID=DISP,
           TYPE=DEC,
           LENGTH=00007,
           POSITION=00138,
           KEY=NO,
           SNAME='DISB SPARES',
           RELATE=NO,
           SIGN=YES,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
*                                                                          000
*          DICTIONARY ADF FLD: T SAIVDIDV                                  000
*          DICTIONARY   FIELD: TCDISB-DIVERS
*
   FIELD   ID=DIDV,
           TYPE=DEC,
           LENGTH=00007,
           POSITION=00146,
           KEY=NO,
           SNAME='DISB DIVERS',
           RELATE=NO,
           SIGN=YES,
           AUDIT=NO,
           CAUDIT=NO,
           MSG=NO
```

```
*************************************************************************
*
*  SEGMENT: SEGID= CY     DATE:   08/14/84    TIME:   13:00:28
*           DICTIONARY ADF SEG: T SAMPCY                              000
*           DICTIONARY SEGMENT: TCCYCCOUNT                            000
*
*************************************************************************
   SEGMENT  ID=CY,
            PARENT=IV,
            NAME=CYCCOUNT,
            LENGTH=00025,
            KEYNAME=CYCLKEY,
            PCBNO=001,
            DBID=PA,
            TRAILER=01,
            SKSEGS=37,
            KASCEND=YES
*
*           DICTIONARY ADF FLD: T SACYKEY                             000
*           DICTIONARY   FIELD: TCCYCLKEY                             000
*
   FIELD    ID=KEY,
            TYPE=ALPHANUM,
            LENGTH=00002,
            POSITION=00001,
            KEY=YES,
            SNAME='20',
            RELATE=NO,
            SIGN=NO,
            AUDIT=NO,
            CAUDIT=NO,
            MSG=NO
*
*           DICTIONARY ADF FLD: T SACYCNTA                            000
*           DICTIONARY   FIELD: TCPHYS-COUNT                          000
*
   FIELD    ID=CNTA,
            TYPE=DEC,
            LENGTH=00007,
            POSITION=00003,
            KEY=NO,
            SNAME='PHYS COUNT',
            RELATE=NO,
            SIGN=YES,
            AUDIT=NO,
            CAUDIT=NO,
            MSG=NO
*
*           DICTIONARY ADF FLD: T SACYSTCK                            000
*           DICTIONARY   FIELD: BOOK-COUNT                            000
*
   FIELD    ID=STCK,
            TYPE=DEC,
            LENGTH=00007,
            POSITION=00011,
            KEY=NO,
            SNAME='BOOK COUNT',
            RELATE=NO,
            SIGN=YES,
            AUDIT=NO,
            CAUDIT=NO,
            MSG=NO
```

```
************************************************************************
*
*  CREATE STATIC RULES - SEGMENT LAYOUT AND SEGMENT HANDLER
*
************************************************************************
   GENERATE OPTION=SGALL
************************************************************************
*
*  CREATE DEFAULT TRANSACTIONS - CONVERSATIONAL ITR AND SCREENS
*
************************************************************************
   GENERATE OPTION=CVALL,
            TRXID=PA,
            DBPATH=PA
   GENERATE OPTION=CVALL,
            TRXID=PD,
            DBPATH=PD
   GENERATE OPTION=CVALL,
            TRXID=IV,
            DBPATH=IV
   GENERATE OPTION=CVALL,
            TRXID=CY,
            DBPATH=CY

************************************************************************
*
*  CREATE OR UPDATE THE CONVERSATIONAL SECONDARY OPTION MENU RULE
*
************************************************************************
   GENERATE OPTION=SOMSS
ADFY031 I PROCESSING SUCCESSFULLY COMPLETED
```

APPENDIX C. ADFIN PROCESSOR MODULE DEFINITION

MCF1Y09	ADFIN Processor Main Routine
MCF1Y10	ADFIN Processor ADFSYS01 Routine
MCF1Y11	ADFIN Processor ADFSEG01 Routine
MCF1Y12	ADFIN Processor ADFDTE01 Routine
MCF1Y03	ADFIN/ADFOUT Message Handler
MCF1Y35	ADFIN/ADFOUT Message Parsing Routine
MCF1Y36	ADFIN/ADFOUT Message Table
MCF1Y37	ADFIN/ADFOUT Message Build Routine
DBDWLNKA	DB/DC Data Dictionary Assembler Interface
MCF1V38	IMSADF II DDNAME Checker
MCF1V39	IMSADF II Trace Interface
MCF1A16	IMSADF II Date/Time Routine
ADFOPTNS	IMSADF II Installation Definition
????Y09	ADFIN Processor link-edit load module name. ???? is the installed ADFID in effect at link-edit time. This is the PROGRAM-NAME entered on the EXECUTE command.

APPENDIX D. ADFOUT PROCESSOR MODULE DEFINITION

MFC1Y01	ADFOUT Processor Main Routine
MFC1Y02	ADFOUT Processor Table Definitions
MFC1Y03	ADFOUT Processor Message Handler
MFC1Y04	ADFOUT processor SYSTEM statement routine
MFC1Y05	ADFOUT processor SEGMENT statement routine
MFC1Y06	ADFOUT processor FIELD statement routine
MFC1Y07	ADFOUT processor GENERATE statement routine
MFC1Y08	ADFOUT processor user-specified data set output routine
DBDWLNKA	DB/DC Data Dictionary Assembler Interface
MFC1Y35	ADFIN/ADFOUT Message parsing routine
MFC1Y36	ADFIN/ADFOUT processors message table
MFC1Y37	ADFIN/ADFOUT message build routine
MFC1V38	IMSADF II DDNAME Checker
MFC1V39	IMSADF II Trace Interface
MFC1V48	IMSADF II Partitioned data set output handler
MFC1A16	IMSADF II Date Time Routine
ADFOPTNS	IMSADF II Installation Definition
????Y01	ADFOUT processor link-edit load module name. ???? is the installed ADFID in effect at link-edit time. This is the PROGRAM-NAME entered on the EXECUTE command.

Note: MFC1V40 is an additional IMSADF II module which is needed only if the ADFTRACE is executed.

APPENDIX E. IMSADF II STRTYPE

While the IMSADF II Data Dictionary Extension does not reference STRTYPE in processing, certain Dictionary commands, such as COPY and EXPORT, require a subject in STRTYPE. A listing of commands is needed to create one such structure type, producing ordered reports against the IMSADF II model. Such a report could be helpful in verifying the accuracy of ADFOUT output in problem situations.

Note: The STRTYPE subject for IMSADF II structures does not have to be named (*,,ADFDDE,0). That name is used for illustrative purposes only.

```
ADD STRTYPE (P,,ADFDDE,0) DATE=840405;

AR  STRTYPE (P,,ADFDDE,0) CONTAINS RELTYPE (*,,ADFSYS01/TO/DBS,0) +
   SEQ=10 SUPERCAT=L FOLLOWDN=Y FOLLOWUP=Y;

AR  STRTYPE (P,,ADFDDE,0) CONTAINS RELTYPE (*,,ADFSYS01/TO/ADFSEG01,0) +
   SEQ=20 SUPERCAT=L FOLLOWDN=N FOLLOWUP=N;

AR  STRTYPE (P,,ADFDDE,0) +
   CONTAINS RELTYPE (*,,DATABASE/WITH/SEGMENT,0) +
   SEQ=30 SUPERCAT=L FOLLOWDN=Y FOLLOWUP=Y;

AR  STRTYPE (P,,ADFDDE,0) CONTAINS RELTYPE (*,,ADFSEG01/TO/SEG,0) +
   SEQ=40 SUPERCAT=R FOLLOWDN=Y FOLLOWUP=Y;

AR  STRTYPE (P,,ADFDDE,0) CONTAINS RELTYPE (*,,ADFSEG01/TO/ADFDTE01,0) +
   SEQ=50 SUPERCAT=L FOLLOWDN=N FOLLOWUP=N;

AR  STRTYPE (P,,ADFDDE,0) CONTAINS +
   RELTYPE (*,,SEGMENT/WITH/ELEMENT,0) +
   SEQ=60 SUPERCAT=L FOLLOWDN=Y FOLLOWUP=Y;

AR  STRTYPE (P,,ADFDDE,0) CONTAINS +
   RELTYPE (*,,ELEMENT/CONTAINS/ELEMENT,0) +
   SEQ=70 SUPERCAT=L FOLLOWDN=Y FOLLOWUP=Y;

AR  STRTYPE (P,,ADFDDE,0) CONTAINS RELTYPE (*,,ADFDTE01/TO/DTE,0) +
   SEQ=80 SUPERCAT=R FOLLOWDN=Y FOLLOWUP=Y;

INSTALL STRTYPE (P,,ADFDDE,0) UPDATE=YES;
```

The STRUCTURE REPORT command used with STRTYPE ADFDDE, or equivalent, may be used to produce a list of IMSADF II fields and segments under an ADFSYS01 subject.

An example of the command and the resulting report follows:

```
STRUCTURE_REPORT ADFSYS01 (T,,SAMP,0) +
 STRUCTURE=ADFDDE AND DEPENDENTS DESC=0 +
 CATRPT=(ADFSEG01,ADFDTE01) +
 DEST=L;

  * * * * * * * *      DB/DC DATA DICTIONARY REPORT      10/19/84 08:43:33
                                                          PAGE:0001

STRUCTURE REPORT FOR:  ADFSYS01 T  SAMP 0
CATEGORY               RC  SUBJECT NAME      SEQATTR      REL KEYWORD
------------------------------------------------------------------------------

ADFSEG01                   T  SAMPPA 0                    ADFSYS_HAS
ADFSEG01                   T  SAMPPD 0                    ADFSYS_HAS
ADFSEG01                   T  SAMPIV 0                    ADFSYS_HAS
ADFSEG01                   T  SAMPCY 0                    ADFSSY_HAS
     ADFSEG01           R  T  SAMPPA 0                    POINT_ADFSEG
       ADFDTE01            T  SAPAKEY 0                   ADFSEG_HAS
       ADFDTE01            T  SAPADESC 0                  ADFSEG_HAS
     ADFSEG01           R  T  SAMPPD 0                    POINT_ADFSEG
       ADFDTE01            T  SAPDKEY 0                   ADFSEF_HAS
       ADFDTE01            T  SAPDPRCD  0                 ADFSEG_HAS
       ADFDTE01            T  SAPDINVC 0                  ADFSEG_HAS
       ADFDTE01            T  SAPDPLRV 0                  ADFSEG_HAS
       ADFDTE01            T  SAPDMKDP 0                  ADFSEG_HAS
       ADFDTE01            T  SAPDCOMM 0                  ADFSEG_HAS
       ADFDTE01            T  SAPDRISP 0                  ADFSEG_HAS
       ADFDTE01            T  SAPDWRSP  0                 ADFSEG_HAS
     ADFSEG01           R  T  SAMPIV 0                    POINT_ADFSEG
       ADFDTE01            T  SAIVW 0                     ADFSEG_HAS
       ADFDTE01            T  SAIVAREA 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVINVD 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVPROJ 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVDIV 0                   ADFSEG_HAS
       ADFDTE01            T  SAIVFILL  0                 ADFSEG_HAS
       ADFDTE01            T  SAIVPRIC 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVRPRI 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVUNIT 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVCOAP 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVPLAN 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVCOAD 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVCDAY 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVTDAY 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVREQC 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVRREQ 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVREQU 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVONOR 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVRONO 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVSTCK 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVRSTC 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVDIPL 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVRDIP 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVDIUN 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVRDIU 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVDISP 0                  ADFSEG_HAS
       ADFDTE01            T  SAIVDIDV 0                  ADFSEG_HAS
     ADFSEG01           R  T  SAMPCY 0                    POINT_ADFSEG
       ADFDTE01            T  SACYKEY 0                   ADFSEG_HAS
       ADFDTE01            T  SACYCNTA 0                  ADFSEG_HAS
       ADFDTE01            T  SACYSTCK 0                  ADFSEG_HAS
------------------------------------------------------------------------------
                         * * * END-OF-REPORT * * *
```

INDEX

IMS Application Development Facility II Version 2 Release 2
Data Dictionary Extension User's Guide
SH20-6597-01

READER'S COMMENT FORM

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

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

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

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

____

____

____

____

What is your occupation? ______________________________

Number of latest Newsletter associated with this publication: ____________________

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

SH20-6597-01

Reader's Comment Form

Cut or Fold Along Line

Fold and tape   **Please Do Not Staple**   Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS   PERMIT NO. 40   ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 8D8
220 Las Colinas Boulevard
Irving, Texas 75039-5513

Fold and tape   **Please Do Not Staple**   Fold and tape

IBM®

Publication Number
SH20-6597-01

File Number
S/370/4300-32

Program Number
5665-348

SH20-6597-01