



400 East Pratt Street  
Baltimore, Maryland 21202 (410) 539-3733

# **AIS+ EE**

## **Advanced Image Solution+ Enterprise Edition**

**Version 1.0**

# **VSR Manual**

# Table Of Contents

---

**COPYRIGHT** ..... 3

**TRADEMARKS**..... 3

**SOFTWARE PREREQUISITES FOR VSR** ..... 3

**NEW FEATURES OF VSR VERSION 3.0** ..... 4

**ABOUT THIS MANUAL**..... 5

**INTRODUCTION**..... 6

1. VALIDATION ..... 6

2. OAM STORE..... 6

3. UPDATE/REGISTRATION ..... 7

USABILITY AND REUSABILITY OF REJECT AND OUTPUT FILES: ..... 7

**PROCESS OVERVIEW**..... 8

CONTROL FILE ..... 8

INPUT FILE (INDATA1) ..... 8

**PROCESSING OF INPUT RECORDS**..... 12

**VSR CONTROL FILE** ..... 13

**VSR INPUT FILE LAYOUTS**..... 24

**OBJECT DISTRIBUTION MANAGER V2.2 AND FAF V2.2 FORMAT**..... 25

LAYOUT OF THE INPUT SEGMENT RECORD FOR V2.2 ..... 36

ALTERNATE INPUT FILE VERSIONS ..... 38

LAYOUT OF THE ALTERNATE INPUT SEGMENT RECORD FOR PRE-V2.2..... 42

**INPUT VALIDATION PROCESSING: FAF TABLE VALIDATION**..... 43

**SPECIAL CONSIDERATIONS: AIS+/FAF TABLES USAGE** ..... 45

**INPUT VALIDATION PROCESSING: CUSTOM VALIDATION PROCESS** ..... 45

**VSR CUSTOMER EXITS**..... 46

**GUIDELINES AND RESTRICTIONS** ..... 46

DATA COLLECTION EXIT: FOR FAF TABLES..... 47

PROGRAM PROCESSING LOGIC..... 54

**CUSTOMER DATA EXIT: FOR FAF TABLES**..... 55

CALL STATEMENT..... 55

PARAMETER LIST STRUCTURE ..... 56

COPY MEMBER AILCSDTA ..... 64

PROGRAM PROCESSING LOGIC..... 68

**ROUTING DATA EXIT: FOR FAF TABLES**..... 69

CALL STATEMENT..... 69

PARAMETER LIST STRUCTURE ..... 69

COPY MEMBER AILROUTE ..... 74

**EVENT FORMAT EXIT: FOR FAF TABLES ..... 76**

CALL STATEMENT ..... 76

DEFAULT SUPPLIED FORMATS ..... 77

PARAMETER LIST STRUCTURE ..... 78

COPY MEMBER AISW013 ..... 81

**CUSTOMER EXIT LOCATIONS AND FUNCTION CODES: FOR FAF TABLES ..... 83**

**FUNCTIONS AND EVENT CODES: FOR FAF TABLES ..... 84**

OFF-LINE EXIT FUNCTION CODES ..... 84

**CUSTOM DATA VALIDATION EXIT: FOR NON-FAF IMPLEMENTATION ..... 85**

CALL STATEMENT ..... 85

PARAMETER LIST STRUCTURE ..... 85

**CUSTOM DATA REGISTRATION EXIT: FOR NON-FAF IMPLEMENTATION ..... 86**

CALL STATEMENT ..... 86

PARAMETER LIST STRUCTURE ..... 86

**VSR ERROR MESSAGES AND RECOVERY ..... 87**

**INSTALLATION OF VSR SOFTWARE ..... 90**

**STORING OBJECTS USING VSR ..... 95**

**APPENDIX A: AISRCOV TABLE ..... 98**

**APPENDIX B: STORAGE OF IMAGE OVERLAYS ..... 103**

FEATURE SET-UP ..... 103

RUNNING THE FORMS OVERLAY STORAGE FEATURE ..... 105

## Copyright

---

The entire contents of this manual are copyright 1995 by SYSCOM, Inc. All rights reserved.

## Trademarks

---

The following trademarks and registered trademarks contained in this manual belong to International Business Machines (IBM).

- DB2 - Database 2
- CICS - Customer Information Control System
- FAF - Folder Access Facility
- ImagePlus
- OAM - Object Access Method

## Software Prerequisites for VSR

---

The following software components and release levels are required for VSR:

- MVS/ESA 3.2 or higher
- DB2 2.3 or higher

## **New Features of VSR Version 3.0**

---

1. VSR 3.0 is year 2000 compliant and includes an additional step in the run JCL and a new program, AICTDRVR. These changes have been outlined in the installation of VSR software section of the manual
  - The run JCL has been modified to add a new step prior to the step that runs the VSR program.
  - The step that executes the VSR program will have to be modified to add a new input DD statement for the temporary file created from the new step identified above. Please review the sample VSR run JCL supplied with the software for more information.
  - A new batch program AICTDRVR will have to be installed.
2. The VSR 3.0 can be now be used to store the objects that are in TIFF format only for ODM V2.2 (12-1995 maintenance level) or higher version databases.

## About This Manual

---

The Validate, Register and Store (VSR) Manual is intended for the user responsible for installing and executing the batch store process of image data and for users responsible for customizing that process.

For more general information (overview) on AIS+ and imaging, please refer to the AIS+ General Information Manual and the appropriate IBM ImagePlus manuals.

## Introduction

---

The VSR process will insert objects generated by the user (either via high-speed scanning, coded data generation, etc.) into the ImagePlus FAF V2.X databases and ODM V2.1, V2.1.1, or V2.2 databases. The process can also be configured to perform validation of input data, and updates to custom databases.

In either case, the VSR facility begins by having objects generated and placed in validation input format (this will be discussed later in this manual). Objects which enter the VSR process via the validation step will be required to have the object data in MO:DCA (Mixed Object Document Content Architecture) format.

The object attributes are grouped based on the following:

- Indexing information - This consists of information necessary for indexing an object in the AIS+ and FAF database, or custom databases through a custom exit.
- Storage Management Specification (SMS) information - This consists of information specific to the storage of the object in the Object Access Management (OAM) database.
- Routing information - This consists of information required for creating a work schedule activity for the object in AIS+ or a custom database.

Only the SMS is present for non-AIS+ validated objects unless user generated custom programs provide additional functionality through the user exits.

The user can specify the values of all the parameters belonging to a sub-group. Alternatively, the user can work with the default parameters for any of the above mentioned sub-groups by using a form number.

Conceptually, the VSR process consists of 3 distinct sub-processes. The three sub-processes are:

### 1. Validation

Validation verifies the correctness of the control and indexing information for the objects specified in the control (CNTRLDAT) and input (INDATA1) files. The validation is performed against the AIS+ and FAF tables, or can be validated against custom tables through the use of user exits. Subsequently, when data is determined to be valid, it produces:

- a valid object member which, based on control flags, will be input to the OAM store process or output to file OUTDATA;
- a reject file (ERRDATA), which can be corrected and resubmitted to the process;
- various reports (ERRRLOG1,ERRRLOG2,WORKRPT1,WORKRPT2), and;
- destination device messages.

### 2. OAM Store

The OAM store process will read run control and object information provided by:

- the Control File (CNTRLDAT), and;
- the validation output object member, or user prepared and supplied object record data.

- This data, coupled with control file data, will provide the OAM Store process with the information needed to insert the object into the OAM. Additionally, it creates: entries in the batch reject file (ERRDATA) which contains OAM Store rejected records, and various reports (ERRRLOG1,ERRRLOG2,WORKRPT1,WORKRPT2).

### 3. Update/Registration

The update/registration process maintains the AIS+ and FAF database (where appropriate) with accurate indexing and routing information about the objects stored in the database(s).

Update reads information provided by the Validate and OAM Store processes and produces:

- reports (ERRRLOG1,ERRRLOG2,WORKRPT1,WORKRPT2) and destination device messages;
- updates to the AIS+ and FAF databases;
- updates to the object table and an event in the event table, and;
- customer exits to allow for custom updates through user provided programs.

#### Usability and Reusability of Reject and Output files:

In the validation process, the layout of the records in the reject files will be the same as the layouts used in the processing of the data. In this fashion the records can be corrected and resubmitted for processing. The user selects the options for processing in the control file (CNTRLDAT). The selection of control options and input files must be carefully controlled to ensure that the proper processing sequences are followed. Records placed in the ERRDATA file are re-sequenced by resetting the Record ID field to a new count based on the number of records written to the reject file. This will allow for range processing should the reject file be reprocessed.

The OAM Store process also creates entries in the file ERRDATA for rejected records. Records are rejected in the store process due to the systems inability to correctly store the record. The format of records in the ERRDATA reject file is based on the input file format. These records can be corrected and reentered into the process providing the user selects the proper control options. The error report will indicate the nature of the store failure.

Failures in the registration process can cause a rollback to the last commit point, an entry into the recovery table, and program termination. The error report detailing the update/registration failure will indicate the nature of the error in the process. Failures in the process that are not related to the validity of the data could also result in program termination. See the section detailing VSR Recovery/Restart for information concerning actions to take should this situation occur.

The Reject files generated in the process may overlay the existing reject file's records which were generated in the previous run. It is recommended for the customer to use Generation Data Groups (GDG's) and/or appropriate JCL DISP parameters to control the use of the files.

The error report(s) generated during the process will detail the reasons for the creation of the reject records. Examination of the error report(s) will assist in the correction of the rejected records.



## Process Overview

---

The VSR process is configured to meet individual customer requirements. Validation can be performed as a stand alone process, or Validation, OAM Store, and Update/Registration can be performed as a combined process. The processing is configured through the setting of parameters in the input control file (CNTRLDAT). Input to the process is the sequential file generated by the unload of the High Speed Capture System (HSCS) VSAM data sets, or by another process that generates the VALIN formatted records. Rejected records generated from the process that have been corrected can also be input. The following input files are required for the process:

### Control File

The Control File will have information global to all the objects being verified, and the processing parameters involved. This file allows for comment records, a keyword or text area, followed by a KEY NUMBER, which precedes the data.

### Input File (INDATA1)

The Input File will have the object data and object attributes. This file will contain two types of records (comment type records, type = 99, are ignored). The first type will be referred to as:

1. The Indexing Record - This record contains the following sub-areas of information, which are described later.

- Base data
- Indexing data
- Routing data
- Return parameters
- Object data

The second type of record within the Input File will be referred to as:

2. The Segment Record - The existence of this record indicates that the object contains multiple records, and this is a continuation of a previously referenced object. The segment record contains:

- Base data (which links the record with a previous indexing record)
- Object data

The organization of the input file is as follows:

#### Indexing Record

The object details (indexing, routing information) are provided in the indexing record. Additionally, the indexing record may also contain the actual object data. It is possible for the user to simply place the base and non-object related portion of the record in the indexing record, while always starting the object data in a segment record. However, nothing precludes the user from placing both the indexing information and part or all of the object data in the indexing record.

The base area within either the indexing or segment record contains general object information, and is repeated in every segment record associated with the indexing record. The indexing area contains the object indexing data. The routing area contains the work scheduling data. The routing parameters will have absolute values. The unit code and route code provided by the user will specify the final destination. The object area contains the details of the object data. The variable entry area contains all the variable data. Thus, each object to be stored will have an indexing record.

The maximum size of the indexing record is 32KB. It is suggested that the object attribute and data be put together in the indexing record if the object attributes and the data are less than 32KB. If the indexing record cannot contain the entire object data, segment records must be created for this object.

#### Segment Record

The segment records are associated with an indexing record. If the actual object data cannot fit completely, or if the indexing record contains only object details, then the actual object data will be provided in the segment record(s). In the case of large objects, the object data will be split across multiple segments. The segment records will contain base data and object data.

Validation creates the following output files (based on control options):

#### Validated Object File (OUTDATA)

The OUTDATA file will have the validated index and segment records from the input file, with some fields created by the validation process for subsequent processing steps. Through control file options to indicate FAF V2.2 record input this file can be processed through the store and update/registration process and the validation process can be bypassed. A custom validation process can be implemented with sample programs provided and the control options set appropriately.

The file records will be formatted the same as the input data sequential file output from the IBM Batch Utility jobs ECKRUNUT and ECKRUNU2.

#### Reject File (ERRDATA)

The Reject file will have the rejected index and segment records from the input file. The file records will be formatted the same as the input file member based on the control file parameters specifying file input formats. For FAF validation, the error report generated from the validation process will indicate what errors triggered the generation of the error records. Once the errors are corrected, the rejected record file can be run through the process, provided the appropriate control file flags are set. If a custom validation process is used then error reporting must be implemented.

Reports containing exceptions which took place, and all of the control information inserted by the user.

The Report consists of four parts:

1. Control Parameter section

This section consists of the values of all the keyword parameters in the control file. When errors are encountered, the errors are listed after the control parameter section.

## 2. Input Parameter section

This section consists of a list of values for the object record parameters. When errors are encountered, the errors are listed after the input parameter section. Optionally, the user can list all the input records in the report, depending on the "Run Control Option" keyword set in the control file.

For the FAF validation records with index/routing data in error, the report will indicate that the record contains index errors which were significant for the store and update/registration process. One will recognize this record to be a record with index/routing data in error by the special return code (+8) and record information detailed in the report. This report could serve as a correcting aid during the on-line process.

If the user chooses to implement a custom validation process through supplied exits, then reporting is limited to output of parameter values. Custom printing of errors could be implemented in the custom exits.

## 3. Run Statistics

This section consists of a list of statistics regarding the validation of the run. It also shows the size of objects processed by validation.

## 4. Job Return Code

This section contains the return code of the job.

Based on control options set in input file CNTRLDAT, the process will bypass the creation of the validated record file and process the records directly through the store and update/registration process. In this case index and segment records rejected in the validation and store/registration process are entered in file ERRDATA to allow for correction and reentry to the process. If the processing is configured to pass the input records through the entire process simultaneously, or if a record is successfully stored but fails in the update/registration process, a rollback to the previous commit point is performed. The process is then halted, and the user must correct the data and restart the process. See the Recovery/Restart description for information concerning this.

The Store and Update/Registration process is performed together for each record. The process creates the following output files (based on control options):

### Reject File (ERRDATA)

The Reject file will have the rejected index and segment records from the input file. The file records will be formatted the same as the input data. The error report generated from the update/registration process will indicate what errors triggered the generation of the error record. Once the errors are corrected, the rejected record file can be run through the process, providing the appropriate control file flags are set.

Reports containing exceptions which took place, and run statistics.

The Report consists of three parts:

### 1. Errors

This section consists of a list of errors encountered in the store and update/registration process.

## 2. Run Statistics

This section consists of a list of statistics regarding the store and update/registration process of the run.

## 3. Job Return Code

This section contains the return code of the job.

## Processing of Input Records

---

Records input to the VSR process can be from ODM V1.2, V2.1.1, or V2.2. Control file options must be set to indicate the type of records being input to the process. VSR reformats the input data into ODM V2.2 record format, which is used in the processing of records internally. Objects are stored in ODM version 2 format and updates are performed to AIS+ and FAF V2.X databases (or custom databases if the custom option is chosen). The input file formats for the control file and object data are presented in the following section.

## VSR Control File

The VSR control file CNTRLDAT contains control options configuring the processing. This information is input into the process and verified as the initial step in the process. Any errors in the control file halt the processing, and require correction and resubmittal of the job.

Processing options entered should be carefully considered. The RUNCNTRL variable enables the user to select the processing flow. The CNTRLDAT in conjunction with a recovery entry can be used to restart the process in the event of a job execution interruption. The following details the fields entered in the file, and provides a sample input file format.

VSR Control File CNTRLDAT				
Keyword	Key Number	Size of Value (Bytes)	Req	Description
BATRUNIT=	01	Char (20)	O	The batch run ID is a 20-character name given by the user to identify the run name. The default is the name generated by the validation process in the form of a DB2 datestamp. This ID, if unique, can be used in the restarting of the process in the event of a failure.
RUNCNTRL=	02	Num (2)	R	The run control option has the following values: 01 Normal run, all records, validation, store, update/registration 02 Validate control file only 03 Print only, all records 04 Print only, all records, validate and print errors 05 Normal run, all records, validation, store, update/registration, record range 06 Print only, record range, 07 Store and update/registration, all records 08 Store and update/registration, record range
STRECORD=	03	Num (9)	O	The start record ID is used when the run control option is 05, 06 or 08. It represents the beginning record in a processing range, and must be an index record. This is equivalent to the Record ID number in the input data sets. This value is overridden in a recovery scenario by the record value of the last processed record in the recovery table AISRCOV. For ODM V1.2 and V2.1 this field must be six characters only, zero filled. For ODM V2.2 this field is nine positions, zero filled if necessary.

ENRECORD=	04	Num (9)	O	The end record ID is used when the run control option is 05, 06 or 08. It represents the last record in a processing range, and must be an index type record. This is equivalent to the Record ID number in the input data sets. For ODM V1.2 and V2.1 this field must be six characters only, zero filled. For ODM V2.2 this field is nine positions, zero filled if necessary. For range processing either ENRECORD or NMOBJECT must be used, with only one value allowed.
-----------	----	------------	---	--

<b>VSR Control File CNTRLDAT</b>				
<b>Keyword</b>	<b>Key Number</b>	<b>Size of Value (Bytes)</b>	<b>Req</b>	<b>Description</b>
NMOBJECT=	05	Num (9)	O	The number of objects is used when the run control option is 05, 06 or 08. It is used instead of the end record to request the number of objects to process, starting at the record number specified in the start record. For range processing either ENRECORD or NMOBJECT must be used, with only one value allowed.
APPLCODE=	06	Num (2)	R	For FAF only: The application code is a unique value to be used in the generation of stored data. This must be a valid value in the AIS+ and FAF tables. Input record application code values must match this, or they will be rejected.
MESGDEST=	07	Num (2)	O	The message destination code identifies the mode of display for error and progress messages. The values are:  01 All messages are sent to the system output (SYSOUT) destination. The default is 01.  02 All messages are sent to the SYSOUT destination and to the console.
MSGCOUNT=	08	Num (3)	O	The progress message count generates a progress message after the number of index records specified are processed. If this value is 0, which is the default, then no progress messages are displayed or printed. The message count is 0 to 999.
ERRTHRES=	09	Num (4)	O	The error threshold value is the number of errors tolerated. Each error encountered in validation of fields increments the error count by one. It is in the range of 0 to 9999. If this value is 0, which is the default, the error threshold processing is turned off.
PRNTPAGE=	10	Num (2)	O	The print page value is in the range of 1 to 99. It represents the number of lines specified to be printed on one page, not including header lines. The default value is 60.
VERSRELS=	11	Char (6)	R	The version and release name value must be 020100 for V2.1, 0201.1 for V2.1.1, or 020200 for V2.2.
ODMCNTRL=	12	Num (2)	R	For FAF: The IODM control ID record value must be 01 or 02. The values are:  01 Use IODM ID from ODMRECID  02 Use IODM ID from the input file.
ODMRECID=	13	Char (4)	O	For FAF: The IODM ID identifies where the object is stored. It is required if the value of ODMCNTRL is 01.



<b>VSR Control File CNTRLDAT</b>				
<b>Keyword</b>	<b>Key Number</b>	<b>Size of Value (Bytes)</b>	<b>Req</b>	<b>Description</b>
DB2SYSID=	14	Char (4)	R	The DB2 subsystem ID identifies where the VSR process runs. VSR currently does not support multiple DB2 subsystems for OAM.
USERINFO=	15	Char (54)	O	The user reference information contains the title information that the user wants to appear in reports.
STRVRELS=	16	Char (6)	R	The version and release of the input file. A value of 0201 1 means that a V2.1 input file is used with a block size of 32,760 bytes. A value of 0102 1 means that a V1.2 input file is used with a block size of 32,754 bytes. A value of 0202, which is the default, means that a V2.2 input file is used with a block size of 32760. The value must be set appropriately based on desired input record format.
V12DATEF=	17	Num (2)	O	The V1.2 date format is required if a V1.2 input file is used. Valid values are from 01 to 28
USEREXIT=	18	Char (1)	O	If the value of the USEREXIT field is "0", then the normal validation and update/registration process against AIS+ and FAF databases will be executed. If the value of the USEREXIT field is "1" then the normal validation and update/registration process will be bypassed, and the Custom exits will be called in the validation and update/registration process. These custom exits will allow the customer to implement validation and update/registration to user defined applications. Only the BATSTV22I index record information will be passed in the exits. Verification of the segment data will be performed as part of the VSR process.
CUSTEXIT=	19	Char (1)	O	If the value of the Custexit field is "0", the Validation process ( <i>AIBS002P</i> ) will call the VSR Data Collection exit ( <i>AIEX008P</i> ). The default value is "0". A value of "1" means that the exit will not be called.

<b>VSR Control File CNTRLDAT</b>				
<b>Keyword</b>	<b>Key Number</b>	<b>Size of Value (Bytes)</b>	<b>Req</b>	<b>Description</b>
REJTFILE=	20	NUM (1)	O	For FAF: If the value of the REJTFILE field is "0" then the normal VSR process is done; rejected documents are written to the reject file. If the value of the REJTFILE field is "1" then the documents with index data in error are passed through to be inserted into the rejection folder during the update step. The default value is 0. Currently, a value of 0 is the only value accepted. (If the USEREXIT option of "1" is chosen, then all objects rejected will be written to the reject file and this option is ignored).
REJTFDID=	21	Char (15)	O	For FAF: If the value of the REJTFILE field is "1" then the documents with index data in error are passed through to be inserted into the rejection folder during the update step. The user has the option of entering the value of the default reject folder. If the user enters the folder it must exist within AIS+/FAF. If a reject folder is not entered and the REJTFILE field value is "1", then the records are passed with existing folder information. If the process of update/registration fails with this information, program execution may be halted. If the USEREXIT option is = "1", then all rejected objects will be written to the reject file and this option is ignored.

<b>VSR Control File CNTRLDAT</b>				
<b>Keyword</b>	<b>Key Number</b>	<b>Size of Value (Bytes)</b>	<b>Req</b>	<b>Description</b>
TIMEFRMT=	22	Char (01)	R	<p>The time format field contains the format used in the language specified for the application. The values are as follows:</p> <ul style="list-style-type: none"> <li>1 12-hour format (hh:mm xx)</li> <li>2 24-hour format (hh:mm )</li> </ul> <p>For 12-hour time, a number from 01 to 12 specifies the hour (hh), 00 to 59 specifies the minutes (mm), and AM specifies a.m. or PM specifies p.m. The colon (:) and space ( ) are required in the time field.</p> <p>For 24-hour time, a number from 00 to 23 specifies the hour (hh) and 00 to 59 specifies the minutes (mm). The colon (:) and three spaces ( ) are required.</p> <p>For example, to specify 10:30 in the evening, the time parameter value is:                      10:30 PM for 12-hour time                      22:30 for 24-hour time.</p> <p>This time format will be used when executing customer exits and will describe the time format used for the custom exit option.</p>
DATEFRMT=	23	Char (01)	R	<p>The date format field contains the format used in the language specified for the application. The values are as follows:</p> <ul style="list-style-type: none"> <li>1 mm/dd/yyyy</li> <li>2 dd/mm/yyyy</li> <li>3 dd.mm.yyyy</li> <li>4 yyyy-mm-dd</li> <li>5 dd-mm-yyyy</li> <li>6 dd mm yyyy.</li> </ul> <p>This date format will be used when executing customer exits and will describe the date format used for the custom exit option. Any manipulation of dates in customer exits must return the date information in the described format(s).</p>

**VSR Control File CNTRLDAT**

Keyword	Key Number	Size of Value (Bytes)	Req	Description
ROUTELOB=	24	Char (06)	O	For FAF: Default RLOB value. This value is used to store documents that are rejected in the update/registration process for invalid routing information. Value must be valid. If the USEREXIT option is = "1" then this value is not used. All rejected items will be written to the object reject file.
TRANATYPE=	25	Char (06)	O	For FAF: Default Routing Transaction Type value. This value is used to store documents that are rejected in the update/registration process for invalid routing information. Value must be valid. If the USEREXIT option is = "1" then this value is not used. All rejected items will be written to the object reject file.
USERPRM1=	26	Char (04)	O	For FAF: Default User Parameter 1 value. This value is used to store documents that are rejected in the update/registration process for invalid routing information. Value must be valid. If the USEREXIT option is = "1" then this value is not used. All rejected items will be written to the object reject file.
ROUTCODE=	27	Char (06)	O	For FAF: Default Route Code value. This value is used to store documents that are rejected in the update/registration process for invalid routing information. Value must be valid. If the USEREXIT option is = "1" then this value is not used. All rejected items will be written to the object reject file.
ROUTUNIT=	28	Num (04)	O	For FAF: Default Route Unit value. This value is used to store documents that are rejected in the update/registration process for invalid routing information. Value must be valid. If the USEREXIT option is = "1" then this value is not used. All rejected items will be written to the object reject file.
AWAKEDOC=	29	Char (01)	O	For FAF: Default value to determine if documents that are routed should perform the marry function for other documents in the folder. Valid values are "0" (do not wake up documents), "1" (wake up all documents in the same folder in routing), or "2" (only wake up documents in the same folder and routing queue). If the USEREXIT option is = "1" then this value is not used. All rejected items will be written to the object reject file.
FLDRPREX=	30	Char (01)	O	For FAF: Specify whether when creating new folders the folder type should be used as part of the folder ID. If the USEREXIT option is = "1" then this value is not used. All rejected items will be written to the object reject file. Valid values are (Y)es or (N)o.

<b>VSR Control File CNTRLDAT</b>				
<b>Keyword</b>	<b>Key Number</b>	<b>Size of Value (Bytes)</b>	<b>Req</b>	<b>Description</b>
FLDRTYPE=	31	Char (02)	O	For FAF: Default folder type if new folder and no folder type is passed. If the USEREXIT option is = "1" then this value is not used. All rejected items will be written to the object reject file.
KEEPRCOV=	32	Char (01)	O	Flag to determine if Recovery table entry for run should be kept at successful completion of a process run. Valid values are "Y", keep the table entry, or "N" delete table entry at the end of the processing run. A "C" is placed in the Recovery table RCOV_JOB_TERM_FL field for successful program executions with return codes < +9.
ROUTEDOC=	33	Char(01)	O	Flag to determine whether routing information is edited. 'Y', edit routing fields, 'N', don't edit routing fields.
LOGEVENT=	34	Char (01)	O	For FAF: Flag to determine if events should be logged for registration into FAF tables. Valid values are "Y", log events (FAF exit AIEX014P is called for custom formatting of event data), or "N", logging of events is not performed. If the USEREXIT option is = "1", then this value is not used and any logging of events must take place in the customer registration exit.
PLANNAME=	35	Char (08)	O	User specified DB2 plan name. If entered then this plan is used for accessing DB2 and the OAM system. If not entered, then the plan name is defaulted to "AIBS001P".
JOBRUNNM=	36	Char (08)	O	Optionally used to pull default values for the VSR run from the AISRCOV table. If no value is provided, the value will be defaulted to 'AIBS001P'.

**Note:**

Column positions:

**Keyword** 1 to 9

**Key Number** 25 to 26 (optional)

**Value** starts in column 27.

Comment records may be imbedded in the control parameter file by specifying an asterisk (\*) in position 1.

The key number begins in column position 25 because the first 24 columns can be used to hold the keyword.

In the Size of Value column, the abbreviations are as follows:

**Char** Alphanumeric

**Num** Numeric.

In the REQ column, the abbreviations are as follows:

**R** Required

**O** Optional. The optional fields remain optional unless a dependent field makes is required. For example, ODMRECID record is required if the value of ODMCNTRL is 01.

A sample CNTRLDAT file follows:

```
* BATCH RUN ID RECORD
BATRUNID= 01BATCH: #1
*
* RUN CONTROL OPTION RECORD
* 01 Normal run, all records, validation, store, update/registration
* 02 Validate control file only
* 03 Print only, all records
* 04 Print only, all records, validate and print errors
* 05 Normal run, all records, validation, store, update/registration, record range
* 06 Print only, record range,
* 07 Store and update/registration, all records
* 08 Store and update/registration, record range
*
RUNCNTRL= 0201
*
* START RECORD ID
STRECORD= 0300000000
*
* END RECORD ID
ENRECORD= 0400000000
*
* NUMBER OF OBJECTS
NMOBJECT= 0500000000
* APPLICATION CODE RECORD
APPLCODE= 0601
*
* MESSAGE DESTINATION CODE RECORD (01=JOB SUMMARY)
* (02=CONSOLE AND JOB SUMMARY)
MESGDEST= 0701
*PROGRESS MESSAGE COUNT RECORD (IF 0 THEN NO MESSAGES ARE DISPLAYED)
MSGCOUNT= 08010
*
* ERROR THRESHOLD RECORD
ERRTHRES= 090025
*
* PRINT PAGE LIMIT
PRNTPAGE= 1056
*
* VERSION AND RELEASE RECORD
VERSRELS= 11020200
*
*
* ODM ID CONTROL RECORD (01 USE ODM ID FROM ODMRECID RECORD)
* (02 USE ODM FROM INPUT FILE)
ODMCNTRL= 1201
*
* ODM ID RECORD
ODMRECID= 13IODM
*
* DB2 SYSTEM ID
DB2SYSID= 14DSN5
*
* USER REFERENCE INFORMATION RECORD
USERINFO= 15USER INFORMATION # 1
*
* STRUCTURE VERSION AND RELEASE
* 0200 2= VERSION 2 RELEASE 2
* 0201 1= VERSION 2 RELEASE 1.1
* 0102 1= VERSION 1 RELEASE 2
*
```

```
*
STRVRELS= 160201
*
* VERSION 1.2 INPUT FILE IS USED
*
* VERSION 1.2 DATE FORMAT
*
* CODE COUNTRY FORMAT REMARK
*
* 01 U.S.A. MM/DD/YY
* 02 ALBANIA YY-MM-DD
* 03 ARGENTINA DD/MM/YY
* 04 AUSTRIA YY-MM-DD
* 05 BELGIUM DD/MM/YY
* 06 BULGARIA YY-MM-DD
* 07 CANADA MM/DD/YY
* 08 CzechoslovakiaYY-MM-DD
* 09 DENMARK YY-MM-DD
* 10 FINLAND DD.MM.YY
* 11 FRANCE DD.MM.YY
* 12 GERMANY (GDR) YY-MM-DD
* 13 GERMANY (FRG) YY-MM-DD
* 14 GREECE DD/MM/YY
* 15 HUNGARY YY-MM-DD
* 16 ITALY DD/MM/YY
* 17 NETHERLANDS DD-MM-YY
* 18 NORWAY DD.MM.YY
* 19 POLAND YY-MM-DD
* 20 PORTUGAL DD-MM-YY
* 21 ROMANIA YY-MM-DD
* 22 SOUTH AFRICA YY-MM-DD
* 23 SPAIN DD/MM/YY
* 24 SWEDEN YY-MM-DD
* 25 SWITZERLAND DD.MM.YY
* 26 TURKEY DD/MM/YY
* 27 U.K. DD/MM/YY
* 28 YUGOSLAVIA YY-MM-DD
V12DATEF= 1701
*
* VSR CUSTOM USER EXITS
* 0=PERFORM NORMAL BACH STORE PROCESSING
* 1=CUSTOM EXITS FOR VALIDATION AND UPDATE/REGISTRATION PERFORMED
*
USEREXIT= 180
*
*
* VSR CUSTOMER EXITS
* 0=PERFORM CUSTOMER EXITS
* 1=BYPASS CUSTOMER EXITS
*
CUSTEXIT= 190
*
*
* VSR REJECTION FOLDER PROCESS FLAG
0=REJECT DOCUMENTS TO REJECT FILE
* 1=REJECT DOCUMENTS TO REJECTION FOLDER
REJTFILE= 200
*
* VSR REJECTION FOLDER ID
REJTFDID= 21REJECTFOLDID
*
* VSR TIME FORMAT
```

```
TIMEFRMT= 221
*
* VSR DATE FORMAT
DATEFRMT= 231
*
* BATCH DEFAULT FAF ROUTING LOB VALUE
ROUTELOB= 24AAAAAA
*
* BATCH DEFAULT FAF TRAN TYPE VALUE
TRANATYPE= 25BBBBBB
*
* BATCH DEFAULT FAF USER PARM 1 VALUE
USERPRM1= 26CCCC
*
* BATCH DEFAULT FAF ROUTE CODE VALUE
ROUTCODE= 27DDDD
*
* BATCH DEFAULT FAF ROUTE UNIT VALUE
ROUTUNIT= 281111
*
* BATCH DEFAULT FAF AWAKE DOC VALUE
AWAKEDOC= 29Y
*
* BATCH DEFAULT FAF FOLDER PREFIX
FLDRPREX= 30N
*
* BATCH DEFAULT FAF FOLDER TYPE
FLDRTYPE= 31AA
*
* KEEP RECOVERY TABLE ENTRY
KEEPRCOV= 32Y
*
* VALIDATE ROUTING FIELDS FLAG
ROUTEDOC= 33Y
*
* BATCH FAF KEEP EVENT FLAG
LOGEVNT= 34Y
*
* OPTIONAL DB2 PLAN NAME
PLANNAME= 35AIBS001P
*
* JOB NAME FOR VSR RUN
JOBRUNNM= 36AIBS001J
*
```



## **VSR Input File Layouts**

---

VSR reads in files from Object Distribution Manager V1.2, V2.1, V2.1.1, and V2.2. Reject records in the rejected record file ERRDATA are written in the version used as input to the process, and can be reprocessed after correction with the appropriate control file options. The following section details the input data copy members and the edits performed in the VSR process.

## Object Distribution Manager V2.2 and FAF V2.2 Format

The following table contains the record layout of the indexing record which contains the base data, indexing data, routing data, return parameters, and object data. AIS+ models the input record formats after IBM copy member EKCBOT1. AIS+ creates multiple versions of this copybook to enable the process to read in various versions of the input file. The ODM V2.2 copybook member is named AIBSV22I.

<b>Structure of the Indexing Record AIBSV22I - for ODM V2.2</b>					
<b>Part</b>	<b>Field Name</b>	<b>Value</b>	<b>Format</b>	<b>Req</b>	<b>Description</b>
Base data	Record ID		Num (9)	R	The record ID is the record sequence number identifying the object.
	Record type	10	Num (2)	R	The record type is 10 for indexing records. A value of 00 or 99 indicates comment records.
	Sorting group		Num (3)	O	This field is used to allow the user a grouping field that can be used within a batch of objects. It is also used as a primary sorting field. All segments related to an object must contain the same sorting group value. This field keeps the comments interspersed based on the associated group.
	Segment		Bin (4)	R	This field should be set to 1 for indexing record and incremented by 1 for subsequent segment records.
	Total segments		Bin (4)	R	This field contains the total number of indexing and segment records. It is 1 if the header record contains all the object data. If the header record contains partial object data, or if the entire object data is provided in the segment records, then this field is greater than 1.
	Object size		Bin (4)	R	This field contains the total length of the object to be stored.

<b>Structure of the Indexing Record AIBSV22I - for ODM V2.2</b>					
<b>Part</b>	<b>Field Name</b>	<b>Value</b>	<b>Format</b>	<b>Req</b>	<b>Description</b>
	Command code	STOR	Char (4)	O	This field indicates the VSR function and is for informational purposes in AIS+.
	Application ID Code		Bin (2)	O	For FAF tables: This field contains the unique code which identifies the application. It must match the APPLID on the control record; otherwise it is rejected. For custom validation and registration processes this value is ignored.
	Folder ID		Char (26)	O	For FAF tables: The folder ID is used for indexing the object. This field is required. If this field is not entered, the object will be rejected. If the entry is invalid, the folder type will be used to generate any necessary information. For custom validation and registration process this value is ignored.
	Folder token		Char (26)	O	For FAF tables: This field contains the internal name of the folder. If not specified, it may be obtained from the Folder table based on the folder ID. If the folder ID is not found in the Folder table, the folder token is generated.
Indexing data	Code page ID		Bin (2)	R	The code page field is not used in VSR.
	IODM ID		Char (4)	R	For FAF tables: This field contains the ID of the Object Distribution Manager where objects are stored. It is required if the IODM control ID record in the parameter file equals 02. An entry is verified against the Symbolic table. The record is rejected if entry is invalid or omitted.

<b>Structure of the Indexing Record AIBSV22I - for ODM V2.2</b>					
<b>Part</b>	<b>Field Name</b>	<b>Value</b>	<b>Format</b>	<b>Req</b>	<b>Description</b>
	Form number		Char (10)	R	For FAF tables: The form number of the document is used to obtain the collection name, storage class, management class, and retention period values from the Form Number table to be used in the Object Access Method storage process and the values for security class, original kept, file tab, and object description to be used in the update process. It is also used to obtain the RLOB and transaction type values for objects which are routed in the update process. The formnum length in AIS+ and FAF V2.0 is 16 bytes. AIS+ VSR does not use this field. See "Special Considerations" for details concerning this.
	Override indexing flag	0 or 1	Char (1)	R	For FAF or AIS+ tables: If this field is 0, parameters for indexing and storage are obtained from the front-end tables, unless specified in the data collection exit. If this field is 1, parameters are obtained from the input record. (parameters are collection name, security class, and tab description)

<b>Structure of the Indexing Record AIBSV22I - for ODM V2.2</b>					
<b>Part</b>	<b>Field Name</b>	<b>Value</b>	<b>Format</b>	<b>Req</b>	<b>Description</b>
	Security class		Char (2)	O	For FAF tables: This field contains the security classification of the object. It is required if the override indexing flag equals 1. The value must fall in the range of 0 --> 99; otherwise the record is rejected.
	Original kept	0 or 1	Char (1)	O	For FAF tables: This field indicates if the paper copy is retained. 0 Paper copy is not retained 1 Paper copy is retained. It is required if the override indexing flag equals 1. If it is incorrect and REJTFILE = 1, (correct this value during the on-line process) then this value will be reset to "1" and passed through and the document will not be rejected.
	File tab		Char (8)	O	For FAF tables: The file tab is a logical grouping of documents under which an object is filed within a folder. It is required if the override indexing flag equals 1. The file tab length in AIS+ and FAF V2.0 is 16 bytes. See "Special Considerations" for details concerning this.
	Collection name		Char (44)	R	For FAF tables: The collection name is the user specified name relating to the storage group of the object in the Object Access Method. It is required if the override indexing flag equals 1. If the override indexing flag equals 0, the Form Number table should have a valid collection name. The record is rejected if the entry is not valid.

Structure of the Indexing Record AIBSV22I - for ODM V2.2					
Part	Field Name	Value	Format	Req	Description
	Storage class		Char (8)	O	For FAF tables: This optional field is used to override the default storage class.
	Mgmt. class		Char (8)	O	This optional field is used to override the default management class for a collection name.
	Retention period		Bin (4)	O	<p>The retention period computes the expiration date of the objects in storage. The values are:</p> <ul style="list-style-type: none"> <li>null, 0, -1 Management class values are used</li> <li>1 to 32,767 Actual number of days used from the document creation date</li> </ul> <p>X'7FFFFFFF' Object must be kept indefinitely.</p> <p>See the <i>IBM MVS/DFP V3.2 Object Access Method Application Programmer's Reference</i> manual for more details.</p> <p>This field is not currently used in AIS+.</p>

<b>Structure of the Indexing Record AIBSV22I - for ODM V2.2</b>					
<b>Part</b>	<b>Field Name</b>	<b>Value</b>	<b>Format</b>	<b>Req</b>	<b>Description</b>
	Document date received		Char (10)	R	For FAF tables: This field contains the date on which the document is received. The date must be in the correct format. If it is incorrect and REJTFILE = 1, (correct this value during the on-line process) then this value will be passed through and the document will not be rejected.
	Document date filed		Char (10)	O	For FAF tables: This field contains the date on which the document is filed. If the document is routed, this field is set to blanks in the update process. If the document is not routed and the field is blank, the system date is used. If the document is not routed and the document date filed field contains a correct date, that date is used. If it is incorrect and REJTFILE = 1, (correct this value during the on-line process) then this value will be passed through and the value defaulted to current date.
	User date		Char (10)	O	For FAF tables: This field contains the user defined date. If it is incorrect and REJTFILE = 1, (correct this value during the on-line process) then this value will be passed through and the document will not be rejected.

<b>Structure of the Indexing Record AIBSV22I - for ODM V2.2</b>					
<b>Part</b>	<b>Field Name</b>	<b>Value</b>	<b>Format</b>	<b>Req</b>	<b>Description</b>
Routing data	Routing flag	0 or 1	Char (1)	R	For FAF tables: The routing flag specifies if work activity is scheduled for the object. The values are: 0 Do not route the object 1 Route the object. If it is incorrect and REJTFILE = 1, (correct this value during the on-line process) then this value will be reset to "0" and passed through and the document will not be rejected.
	Override routing flag	0 or 1	Char (1)	O	For FAF tables: The override routing flag is required if the routing flag equals 1. 0 The RLOB and transaction type values are accessed based on the form number in the Form Number table during the update process. 1 The rest of the fields in the routing data section are used from the input file during the update process. If it is incorrect and REJTFILE = 1, (correct this value during the on-line process) then this value will be reset to "0" and passed through and the document will not be rejected.
	User parameter 1		Char (4)	O	For FAF tables: This field can be used for the region code. It is required if the override routing flag equals 1.



<b>Structure of the Indexing Record AIBSV22I - for ODM V2.2</b>					
<b>Part</b>	<b>Field Name</b>	<b>Value</b>	<b>Format</b>	<b>Req</b>	<b>Description</b>
	User parameter 2		Bin (4)	O	For FAF tables: This field can be used for the location code. It is required if the override routing flag equals 1.
	RLOB		Char (6)	O	For FAF tables: This field contains the RLOB. It is required if the override routing flag equals 1.
	Transaction type		Char (6)	O	For FAF tables: This field contains the transaction type. It is required if the override routing flag equals 1.
	Override priority indicator	0/1/2/3 or blank	Char (1)	R	For FAF tables: Override priority indicator is used to select a normal (blank or 0), low (1), medium (2), or high (3) priority for the unit of work. If it is incorrect and REJTFILE = 1, (correct this value during the on-line process) then this value will be reset to "0" and passed through and the document will not be rejected.
	Priority		Bin (2)	N	For FAF tables: The priority is calculated based on the override priority indicator set by the user.
	Assigned user ID		Char (8)	O	For FAF tables: This field indicates the user ID associated with the routing event.
	Hold date		Char (10)	O	For FAF tables: The hold date is the date after which a user cannot work on the document. If it is entered, it must be in the correct format, otherwise the record is rejected.

<b>Structure of the Indexing Record AIBSV22I - for ODM V2.2</b>					
<b>Part</b>	<b>Field Name</b>	<b>Value</b>	<b>Format</b>	<b>Req</b>	<b>Description</b>
	Hold time		Char (8)	O	For FAF tables: The hold time is the time specified after which a user cannot work on the document. The time formats are: 12-hour format hh:mm a 24-hour format hh:mm The hold time format is dependent on the time format value in the Application Profile table. If the time is entered but not the date, the record is rejected. If the date is entered but not the time, it defaults to 12:00 a.m.
	Unit code		Bin (4)	O	For FAF tables: The assigned unit code is required if the override routing flag equals 1.
	Route code		Char (6)	O	For FAF tables: The route code for a work queue is required if the override routing flag equals 1.
	Aging date		Char (10)	O	For FAF tables: The aging date is the date on which the object is put on the workflow queue.
Return parameters	Return code		Num (2)	N	This field contains the return code associated with the object.
	Message code		Char (8)	N	The message code further qualifies the return code of the object. This field is not used in VSR.
Object data	Object class		Char (2)	R	The object class is the object type representation within the object to be processed. It must contain a value of either X'4000', or X'0080' or X'8000'; otherwise the record is rejected.
	Number of pages		Bin (2)	R	This field contains the number of pages in the object.

<b>Structure of the Indexing Record AIBSV22I - for ODM V2.2</b>					
<b>Part</b>	<b>Field Name</b>	<b>Value</b>	<b>Format</b>	<b>Req</b>	<b>Description</b>
	Event control flag	0 or 1	Char (1)	R	For FAF tables: The event control flag has the following values: 0 Event history is not updated 1 Event history is updated.
	Object description length		Bin (2)	R	This field contains the length of the object description. If there is no text for the object description, this field must be 0, otherwise the record is rejected.
	Notes Length		Bin (2)	R	This field contains the length of the object notes. If there is no text for the object notes, this field must be 0, otherwise the record is rejected.
	User data length		Bin (2)	R	This field contains the length of the user data. If there is no text for the user data, this field must be 0, otherwise the record is rejected.
	Object data length		Bin (4)	R	This field contains the length of the object data. If the object data is provided only in the segment record, this field must be 0, otherwise the record is rejected.
	Object description		Char (253)	O	The object description text is required if the object description length is greater than 0.
	Notes		Char (253)	R	For FAF tables: The object notes field is required for AIS+. The user must customize the user exit provided in the HSCS to provide information necessary for AIS+ processing. See " <i>Special Considerations</i> " for information concerning this.
	User data		Char (253)	O	The user text data is required if the user data length is greater than 0. It is stored in the user data field in the Object table.

Structure of the Indexing Record AIBSV22I - for ODM V2.2					
Part	Field Name	Value	Format	Req	Description
	Object data	Actual object data	Char (31581)	O	The object data is in the MO:DCA format. All of the object data can be placed in the indexing record or split between the indexing and segment records. It is required if the length of the object data is greater than 0.

Note:

In the Format column, the abbreviations are as follows:

**Bin** - Binary

**Char** - Alphanumeric

**Num** - Numeric.

In the Req column, the abbreviations are as follows:

**R** - Required

**O** - Optional. The optional fields remain optional unless a dependent field makes it required. For example, the folder ID is required if the folder token is not specified.

**N** - Not required.

## Layout of the Input Segment Record for V2.2

The segment records can hold base data and object data. If the object data is larger than the size of the indexing record or the indexing record contains only object details, the object data is provided in the segment records.

A segment record has a maximum size of 32,752 bytes. Objects that are larger than 31,691 bytes in the indexing record must be separated into multiple segments for processing.

The following table contains the record layout of the segment record. The AIS+ copy member is AIBSV22S.

<b>Structure of the Segment Record AIBSV22S for ODM V2.2</b>					
<b>Part</b>	<b>Field Name</b>	<b>Value</b>	<b>Format</b>	<b>Req</b>	<b>Description</b>
Base data	Record ID		Num (9)	R	The record ID is the record sequence number identifying the object.
	Record type	15	Num (2)	R	The record type is 15 for segment records. A value of 00 or 99 indicates comment records.
	Sorting group		Num (3)	O	This field is used to allow the user a grouping field that can be used within a batch of objects. It is also used as a primary sorting field. All segments related to an object must contain the same sorting group value. This field keeps the comments interspersed based on the associated group. This field is required only if the sorting group field in the indexing record has a value.
	Segment number		Bin (4)	R	This field should be set to 1 for the indexing record and incremented by 1 for subsequent segment records.
	Total segments		Bin (4)	R	This field contains the total number of indexing and segment records. If the header record contains partial object data, or if the entire object data is provided in the segment records, then this field is greater than 1.
	Object size		Bin (4)	R	This field contains the total length of the object to be stored.
	Command code	STOR	Char (4)	O	This field indicates the VSR function and is for informational purposes only.
	Application ID code		Bin (2)	O	For FAF tables: This field contains the unique code which identifies the application.

Structure of the Segment Record AIBSV22S for ODM V2.2					
Part	Field Name	Value	Format	Req	Description
	Folder ID		Char (26)	O	For FAF tables: The folder ID is used for indexing the object. This field is required if the folder token is not specified. The value must be equal to the value specified in the indexing record.
	Folder token		Char (26)	O	For FAF tables: This field contains the internal name of the folder. The value must be equal to the value specified in the indexing record. This field is required if the folder ID is not specified.
Object data	Object data length		Bin (4)	R	This field contains the length of the object data in the segment record.
	Object data	Object Data	Char (31581)	R	

**Note:**

In the Format column, the abbreviations are as follows:

**Bin** - Binary

**Char** - Alphanumeric

**Num** - Numeric.

In the Req column, the abbreviations are as follows:

**R** - Required

**O** - Optional. The optional fields remain optional unless a dependent field makes it required. For example, the folder ID is required if the folder token is not specified.

### Alternate Input File Versions

AIS+ will accept ODM V1.2 and V2.1.1 input files also. The following details the layout of these records. The versions contains similar base, index, routing, and object data. The differences between the inputs are resolved internally in the VSR process. AIS+ VSR also uses copy members AIBSV12I, AIBSV12S, AIBSV21I, and AIBSV21S to process input data.

User entered parameters in the control file CNTRLDAT indicate what version of the input file is being used. The following tables present the general layout of the index and segment layouts for earlier ODM input file versions.

<b>File Index Record - ODM pre Version 2 Release 2 format example</b>		
<b>Field Name</b>	<b>Type/ Length</b>	<b>Description</b>
Record ID	Num (6)	This sequence number identifies the object.
Record type	Num (2)	For this header record, the record type is 10.
Segment number	Bin (2)	This is the number of this segment in this object. For the header record, the segment number is 1.
Total segments	Bin (2)	This is the total number of segments in this object. This is an integer greater than 0.
Folder ID	Char (15)	For FAF: You defined this field in Batch ID table 1 (BATCHTBL) to meet the requirements of your enterprise. AIS+ and IPFAF uses this field as an identification of the folder into which this object is placed.
Object size	Bin (4)	This is the total number of bytes in this object. The object size is an integer greater than 0.
Command code	Char (2)	The value of this field is ST, and is optional for AIS+.
Object name	Char (40)	This is the name of the object.
ODM System ID	Char (4)	Host symbolic identifier.
Form name	Char (10)	You defined this field in Batch ID table 1 (BATCHTBL) to meet the requirements of your enterprise. AIS+/VSR will use the value in this field if present. The user may also use an alternate field provided in the user notes field because of field length differences in FAF versions. See " <i>Special Considerations</i> " for details.

<b>File Index Record - ODM pre Version 2 Release 2 format example</b>		
<b>Field Name</b>	<b>Type/ Length</b>	<b>Description</b>
File tab	Char (3)	You defined this field in Batch ID table (BATCHTBL) to meet the requirements of your enterprise. AIS+/VSR will use the value in this field if present. The user may also use an alternate field provided in the user notes field because of field length differences in FAF versions. See " <i>Special Considerations</i> " for details.
Collection name	Char (40)	This is the collection name for this object. The collection name identifies the storage group, management class, and storage class required by the Object Access Method to store and retrieve this object. This field only contains the US English characters A-Z, 0-9, @, #, and \$.
Override storage class	Char (8)	This is an optional field used to override the default storage class. The storage class defines the level of performance and availability of this object. See the <i>IBM MBS/DFP Object Access Method: Planning, Installation, and Storage Administration Guide</i> for more information.
Override management class	Char (8)	This is an optional field used to override the default Management class. The management class defines the backup, retention, and class transition characteristics of this object. See the <i>IBM MBS/DFP Object Access Method: Planning, Installation, and Storage Administration Guide</i> for more information.
Document receive date	Char (6)	This is the date on which the document is received. This information can be derived from the SCANDATE parameter in the STOREB command. It can also be mapped from a scanner data field. The format is <i>MMDDYY</i> .
Document file date	Char (6)	This is the date on which the document is filed. If blank, the document is not filed.
Paper retention flag	Char (1)	For FAF: You defined this field in Batch ID table 1 (BATCHTBL) to meet the requirements of your enterprise. AIS+/VSR and IPFAF use this field to indicate whether the paper document is retained for this object.
Region code	Char (3)	For FAF: You defined this field in Batch ID table 1 (BATCHTBL) to meet the requirements of your enterprise. AIS+/VSR uses this field as user parameter 1 in FAF version 2.



<b>File Index Record - ODM pre Version 2 Release 2 format example</b>		
<b>Field Name</b>	<b>Type/ Length</b>	<b>Description</b>
Location code	Char (3)	For FAF: You defined this field in Batch ID table (BATCHTBL) to meet the requirements of your enterprise. AIS+/VSR uses this field as user parameter 2 in FAF version 2.
Route flag	Char (1)	For FAF: You defined this flag in Batch ID table (BATCHTBL) to meet the requirements of your enterprise. AIS+ uses this flag as a route flag to determine whether work activity is scheduled for this object.
Routing Line-of-Business (RLOB)	Char (5)	For FAF: You defined this field in Batch ID table (BATCHTBL) to meet the requirements of your enterprise. AIS+/VSR uses this field as a RLOB code to identify the type of business to which this object is related.
Transaction type	Char (5)	For FAF: You defined this field in Batch ID table (BATCHTBL) to meet the requirements of your enterprise. AIS+/VSR uses this field as a transaction type to specify what type of work needs to be done for this object.
Unit code	Num (4)	For FAF: You defined this field in Batch ID table (BATCHTBL) to meet the requirements of your enterprise. AIS+/VSR uses this field as a unit code to identify a department, section, or work area to which this object is routed.
Route code	Char (6)	For FAF: You defined this field in Batch ID table (BATCHTBL) to meet the requirements of your enterprise. AIS+/VSR uses this field as a route code to identify how work is distributed within a unit of the enterprise. This field is used in conjunction with the unit code to determine the routing queue that the object is placed in.
Priority	Num (3)	For FAF: You defined this field in Batch ID table (BATCHTBL) to meet the requirements of your enterprise. AIS+/VSR uses this field as a priority code.
Object type	Char (1)	This field identifies the type of object. The value is one of the object type values you defined in the Object Characteristics table (IDOBJCTB). The following are examples of object types AIS+ VSR will accept. <b>I</b> Image <b>J</b> Overlay image <b>C</b> Coded data AIS+ translates this value to Object Class for ODM version 2.

<b>File Index Record - ODM pre Version 2 Release 2 format example</b>		
<b>Field Name</b>	<b>Type/ Length</b>	<b>Description</b>
Number of pages	Bin (2)	This is the number of pages in this object. It is an integer greater than 0.
Retention period	Bin (4)	This is the number of days the system retains this object. When the retention period is exceeded based on the creation date, the object is scheduled for deletion by the Object Access Method. See the <i>IBM MVS/DFP Object Access Method: Application Programmer's Reference Manual</i> for details. This value is not used for the store process.
Description length	Bin (2)	This is the number of bytes in the description field (DESCRIPTION). The value of this field is from 0 to 60. A value of 0 indicates there is no data in the description field for this object.
Description	Char (60)	This field contains the text of any description for this object. If a description is present, the Object Distribution Manager places the length of the description in the DESCR-LENGTH field.
Notes length	Bin (2)	This is the number of bytes in the notes field (NOTES). The value of this field is from 0 to 120. A value of 0 indicates there is not data in the notes field for this object. AIS+ requires data in this field. See description below.
Notes	Char (120)	This field contains the text of any notes for this object. If notes are present, the Object Distribution Manager places the length of the notes in the notes length field. For FAF tables: The object notes is required for AIS+. The user must customize the user exit provided in the HSCS to provide information necessary for AIS+ processing. See " <i>Special Considerations</i> " for information concerning this.
Error message	Char (80)	<i>Message text.</i>
Object data length	Bin (2)	This is the number of bytes in the object data field (OBJECT DATA). It is the number of bytes in the first or only segment of this object. The value is greater than 0 and less than or equal to 31581.
Object data	Char (32284)	This field contains the first or only segment of this object. Version dependent Note: This field is not part of the copybook. You must define this field.

## Layout of the Alternate Input Segment Record for pre-V2.2

The segment records can hold base data and object data. If the object data is larger than the size of the indexing record or the indexing record contains only object details, the object data is provided in the segment records.

A segment record has a maximum size of 32,752 bytes. Objects that are larger than 31,691 bytes in the indexing record must be separated into multiple segments for processing.

The following table contains the record layout of the segment record. The AIS+ copy members are AIBSV21S and AIBSV12S.

<b>File Segment Record - Pre ODM Version 2 Release 2 example</b>		
<b>Field Name</b>	<b>Type/ Length</b>	<b>Description</b>
Record ID	Num (6)	This sequence number identifies the object.
Record type	Num (2)	For this header record, the record type is 20.
Segment number	Bin (2)	This is the number of this segment in this object. For the header record, the segment number is 1.
Total segments	Bin (2)	This is the total number of segments in this object. This is an integer greater than 0.
Folder ID	Char (15)	For FAF: You defined this field in Batch ID table 1 (BATCHTBL) to meet the requirements of your enterprise. AIS+/VSR and FAF uses this field as an identification of the folder into which this object is placed.
Object size	Bin (4)	This is the total number of bytes in this object. The object size is an integer greater than 0.
Command code	Char (2)	The value of this field is ST, and is not used by VSR.
Object data length	Bin (2)	This is the number of bytes in the object data field (OBJECT DATA). It is the number of bytes in this object segment. The value is greater than 0 and less than or equal to 31581.
Object data	Char (32711)	This field contains an object segment and is version dependent.

## **Input Validation Processing: FAF Table Validation**

---

The input file encompasses two types of records: index and segment. The index records are divided into five distinct areas including: base data, indexing data, routing data, return parameters, and object data areas. The segment records contain only base data and object data areas. These areas are further broken down into fields, each having unique edits. Included in this section will be a high level description of those edits.

A blanket edit is incorporated that simply requires all numeric fields to contain numeric data. It is assumed that all data coming into the Validation Process will comply with this requirement as a result of preliminary editing. Please refer to the preceding layout of the input file to help acquaint yourself with the fields. Please note that not every field that exists on the input file is mentioned in this section, only those fields which are being edited.

The APPLICATION ID CODE is a required field. It must have the same value as the APPLICATION ID CODE field located on the Control file. Any record that fails this edit will be rejected.

The FOLDER ID is a required field. This field will be verified against the Folder Table. If the FOLDER ID is not found out on the table, a warning message will be printed out on the report if the validate only run control option (RUNCNTRL = 04) is selected. If the FOLDER ID is not supplied with a value the record will be rejected.

The IODM ID or OSYSID field is required if the ODMCNTL field on the control file equals 2; otherwise this field is not edited. In such a case, it will be verified against the Symbolic Table. This record will be rejected if the field was either not entered, or not found on the table.

The FORM NUMBER field is required. This value will be verified against the AIS+ Document Table (AISDCMT). The record will be rejected if the value was either omitted or not found on the table.

The OVERRIDE INDEXING FLAG field is required. It must have a value of either 0 or 1. The record will be rejected if an appropriate value does not exist. A value of "1" instructs the Validation program to use the Indexing information supplied on the input record. A value of "0" instructs the Validation program to retrieve all necessary Indexing information from the appropriate tables. Any errors encountered during this process will also cause the record to be rejected.

The SECURITY CLASS field is required if the OVERRIDE INDEXING FLAG equals "1". This field must have a value within the range of 0 and 99. The record will be rejected if it does not fall within this range.

The PAPER RETAIN or ORIGINAL KEPT field is required if the OVERRIDE INDEXING FLAG equals "1". This field must have a value of zero or one. If it does not, a warning message will appear on the report and the value will be defaulted to "1". The record will then be processed.

The FILE TAB field is required if the OVERRIDE INDEXING FLAG equals "1". This value will be verified against the AISTABS table. The record will be rejected if the value is not found on the table.

The COLLECTION NAME field is required if the OVERRIDE INDEXING FLAG equals "1". This value will be verified against the Collection table. The record will be rejected if the value is not found on the table.

The RECEIVED DATE field is required. It must be in CCYY-MM-DD format (V1.2 records are in the date format indicated in the input control file). The record will be rejected if this condition is not satisfied.

The USER DATE field is required. It must be in CCYY-MM-DD format (V1.2 records are in the date format indicated in the input control file). The record will be rejected if this condition is not satisfied.

The ROUTE FLAG field is required. It must contain a value of either "0" or "1". A warning message will appear on the report if this condition is not met when the validation process is performed. If the value is invalid a "0" will be added and the object not routed. The record will then be processed.

The OVERRIDE ROUTING FLAG field is required if the ROUTE FLAG equals "1". In this scenario, it must contain a value of either "0" or "1". A warning message will appear on the report if this condition is not satisfied, and a value of "0" substituted. The record will then be processed using default routing information.

The OVERRIDE PRIORITY field is required. It must contain a value of either spaces, "0", "1", "2" or "3". A warning message will appear on the report if this condition is not satisfied and a value of "0" substituted. The record will then be processed with a priority level of normal.

The HOLD DATE field is optional. If supplied, it must be in CCYY-MM-DD (V1.2 records are in the date format indicated in the input control file) format. The record will be rejected if this condition is not satisfied. If this field is supplied and the HOLD TIME is not, the HOLD TIME will default to 12:00 a.m.

The HOLD TIME field is optional. If supplied, it must be in HH.MM.SS format. The record will be rejected if this condition is not satisfied. If this field is supplied and the HOLD DATE is not, the record will be rejected.

The OBJECT CLASS field is required. It must contain a value of either X'4000', X'0080' or X'8000'. The record will be rejected if this condition is not satisfied.

The EVENT CONTROL FLAG field is required. It must contain a value of either "0" or "1". The record will be rejected if this condition is not satisfied.

The following fields must equal "0" if their corresponding fields are blank. The record will be rejected if this condition is not met.

DESCRIPTION LENGTH

NOTES LENGTH

USER DATA LENGTH

USED LENGTH

**\*\*\* Note: When referring to the NOTES field please be advised that the first 34 bytes are being used to hold the FOLDER TYPE, FORM NUMBER and FILE TAB for AIS+ tables. As a result, the NOTES field will never be equal to spaces. If these fields are not present, then the values in the individual fields designated for these values will be used. This requirement is present due to the different data lengths for these fields in the VALIN record verses the FAF databases.**

## Special Considerations: AIS+/FAF Tables Usage

---

Because of implementation differences between ODM V1.0, ODM V2.0, FAF V1.0, FAF V2.0, and AIS+ implementation, the user can customize the functionality of the user exit EKCBCEXIT in the HSCS. The user can place information concerning the folder type, formnum, and filetab in the object notes field. The folder type is a two byte field used to identify folder default information within AIS+. If this information is not supplied then the default folder type input in the control file is used in the creation of new folders. The formnum length in AIS+ and FAF Version 2 is sixteen bytes. To allow the customer the flexibility of utilizing the entire field, and avoiding the 10 byte limitation of the EKCBCOT1 copybook, a sixteen byte formnum field is placed in notes. The file tab length in AIS+ is sixteen bytes. To allow the customer the flexibility of utilizing the entire field, and avoiding the 8 byte limitation of the EKCBCOT1 copybook, a sixteen byte filetab field should be placed in notes. The notes field should be defined as follows:

Bytes 1 - 2 - Folder type from AIS+

Bytes 3 - 18 - Formnum from AIS+

Bytes 19 - 34 - Filetab from AIS+

Bytes 35 - 253 - Notes

AIS+ VSR will utilize these values in the validation and update/registration process for creating the necessary database entries into AIS+ and FAF. If these values are not supplied, then the original field values supplied will be used in the validation and registration process. In all cases the first 34 bytes of the notes field will be used by the VSR process. Any data placed in these positions will be used for the fields indicated.

## Input Validation Processing: Custom Validation Process

---

The input file encompasses two types of records: index and segment, based on ODM VALIN record format(s). A blanket edit is incorporated that simply requires all numeric fields to contain numeric data. It is assumed that all data coming into the Validation Process will comply with this requirement as a result of preliminary editing. Any editing beyond numeric checking for the validity of the actual values in the numeric fields, and other input values, must be checked in the customer modifiable Data Collection exit to assure validity.

## VSR Customer Exits

---

The VSR Process provides three customer exits for validation, and two user exits and control options to enable the customer to perform external validation and registration. If the client elects to perform external validation and registration the Data Collection, Customer Data, and Routing exits are not performed. During the Validation process the Data Collection exit is performed. The Customer Data and Route exits are performed during the Registration process. The exits are performed based on CUSTEXIT and USEREXIT parameters set in the CNTRLDAT file. Refer to the write-up on each specific exit to determine the affect of the aforementioned parameters.

VSR provides stub programs for user customization and provides the data format that AIS+ passes to each program stub. The names of the customer exit programs provided with AIS+ follow:

- Routing Data - AIEX006P
- Customer Data - AIEX007P
- Data Collection - AIEX008P
- Custom Data Validation - AIEX009P
- Custom Data Registration - AIEX010P
- Event Format - AIEX014P

## Guidelines and Restrictions

---

The following guidelines apply when you customize a user exit:

- Determine what you need to do in your user exits.

- Always use the AIS+ provided customer exit copybooks to ensure compatibility with the AIS+ application.

- Create a new DB2 plan to include the DBRMs when the user exit issues a DB2 call.

When you customize a user exit, do not allow it to:

- Perform any database commits or rollbacks

If you update a database from a user exit and AIS+/VSR facility encounters a problem, the AIS+ facility may not rollback your change. In the case where a user exit encounters a DB2 deadlock or rollback the exit should set the exit return code to a value of 20, and restart logic will be performed under VSR control.

### Data Collection Exit: For FAF Tables

The Validation process (*AIBS002P*) calls the VSR Data Collection exit (*AIEX008P*) when the values of both the customer exit record (CUSTEXIT) and the user exit record (USEREXIT) in the control file (CNTRLDAT) are set to "0". The VSR Data Collection exit can override some of the fields in the validation input file (INDATA1). See the following table for the parameters that can be overridden by the VSR Data collection exit.

The exit is called before the validation of the input fields in the INDATA1 file. Therefore the values provided in the exit are validated using the normal validation process. The ODM/FAF V1R2 and V2R1 input records are reformatted to the V2R2 record type before the exit is called. If you use a V1R2 or V2R1 input file, some of the parameters that are defaulted during reformatting can be overridden also.

Parameters Overridden by VSR Data Collection Exit			
Base Data	Indexing Data	Routing Data	Object Data
Sorting group	Code page ID	Routing flag	Number of pages
Folder ID	IODM ID	Override routing flag	Event control flag
Folder token	Form number	User parameter1	Object description length
	Override indexing flag	User parameter 2	Length of notes
	Security class	RLOB	User data length
	Original kept	Transaction type	Object description
	File tab	Override priority indicator	Notes
	Collection name	Priority	User data
	Override storage class	Assigned user ID	
	Override management class	Hold date	
	Retention period	Hold time	
	Date received	Unit code	
	Date filed	Route	
	User date	Aging date	

**Note:**  
When using V1R2 and V2R1 input files, the object name uses version V2R2 naming conventions for storing and registering objects.



The parameters that cannot be overridden by the VSR Data Collection exit are provided in the table below.

<b>Parameters Not Overridden by VSR Data Collection Exit</b>	
<b>Base Data</b>	<b>Object Data</b>
Record ID	Object class
Segment number	Length of object data
Total segments	Object data
Object size	
Command code	
Application ID code	

When the segment records are validated, the following variables are given the indexing record values if they are not the same:

Folder ID

**Note:** You may customize the exit before running the VSR process, or run the exit as a stub. You may use the sample user exit provided.

Call Statement

The call statement used for the VSR Data Collection exit is as follows:

```
CALL( VSR Data Collection exit ) USING
      VSR Data Collection
ON EXCEPTION
      exception process
END-CALL.
```

Parameter List Structure

The below table contains the fields passed to the VSR Data collection exit. It provides the format of the fields and specifies if the field is input or output or both.

<b>Parameter Description for the VSR Data Collection Exit</b>		
<b>Field Name</b>	<b>Format</b>	<b>Input/Output</b>
Sorting group	Num (3)	I/O (Not used by VSR)
Sorting group flag	Char (1)	O (Not used by VSR)
Folder ID	Char (26)	I/O
Folder ID flag	Char (1)	O

<b>Parameter Description for the VSR Data Collection Exit</b>		
<b>Field Name</b>	<b>Format</b>	<b>Input/Output</b>
Folder token	Char (26)	I/O
Folder token flag	Char (1)	O
Code page ID	Bin (4)	I/O (Defaulted from application table, not available for user change)
Code page ID flag	Char (1)	O (defaulted from application table, not available for user change)
IODM ID	Char (4)	I/O
IODM ID flag	Char (1)	O
Form number	Char (16)	I/O
Form number flag	Char (1)	O
Override indexing flag	Char (1)	I/O
Override indexing pass flag	Char (1)	O
Security class	Char (2)	I/O
Security class flag	Char (1)	O
Original kept	Char (1)	I/O
Original kept flag	Char (1)	O
File tab	Char (16)	I/O
File tab flag	Char (1)	O
Collection name	Char (44)	I/O
Collection name flag	Char (1)	O
Storage class	Char (8)	I/O
Storage class flag	Char (1)	O
Management class	Char (8)	I/O
Management class flag	Char (1)	O
Retention period	Bin (4)	I/O
Retention period flag	Char (1)	O
Date received	Char (10)	I/O
Date received flag	Char (1)	O
Date filed	Char (10)	I/O

<b>Parameter Description for the VSR Data Collection Exit</b>		
<b>Field Name</b>	<b>Format</b>	<b>Input/Output</b>
Date filed flag	Char (1)	O
User date	Char (10)	I/O
User date flag	Char (1)	O
Routing flag	Char (1)	I/O
Routing pass flag	Char (1)	O
Override routing	Char (1)	I/O
Override routing flag	Char (1)	O
User parameter 1	Char (4)	I/O
User parameter 1 flag	Char (1)	O
User parameter 2	Bin (4)	I/O
User parameter 2 flag	Char (1)	O
RLOB	Char (6)	I/O
RLOB flag	Char (1)	O
Transaction type	Char (6)	I/O
Transaction type flag	Char (1)	O
Override priority indicator	Char (1)	I/O
Override priority indicator flag	Char (1)	O
Priority	Bin (2)	I/O
Priority flag	Char (1)	O
Assigned user ID	Char (8)	I/O
Assigned user ID flag	Char (1)	O
Hold date	Char (10)	I/O
Hold date flag	Char (1)	O
Hold time	Char (8)	I/O
Hold time flag	Char (1)	O
Unit code	Bin (4)	I/O
Unit code flag	Char (1)	O
Route code	Char (6)	I/O

<b>Parameter Description for the VSR Data Collection Exit</b>		
<b>Field Name</b>	<b>Format</b>	<b>Input/Output</b>
Route code flag	Char (1)	O
Aging date	Char (10)	I/O
Aging date flag	Char (1)	O
Number of pages	Bin (2)	I/O
Number of pages flag	Char (1)	O
Event control flag	Char (1)	I/O
Event control pass flag	Char (1)	O
Object description length	Bin (2)	I/O
Notes length	Bin (2)	I/O
User data length	Bin (2)	I/O
Object description	Char (253)	I/O
Object description flag	Char (1)	O
Notes	Char (253)	I/O
Notes flag	Char (1)	O
User data	Char (253)	I/O
User data flag	Char (1)	O
Object name	Char (40)	I/O
Object name flag	Char (1)	O
Object Class	Char (2)	I
Date format	Char (01)	I
Time format	Char (01)	I
Return code	Num (2)	I/O

**Note:**

In the Format column, the abbreviations are as follows:

**Bin** - Binary

**Char** - Alphanumeric

**Num** - Numeric

In the Input/Output column, the abbreviations are as follows:

**I/O** - The field is input and output and the user can modify the field.

**I** - The field is only input and the user cannot modify the field.

**O** - The field is only output and the user can modify the field.

Copy member AIBSDTCL is used for the call to the Data Collection program AIEX008P. The member follows:

Copy member AIBSDTCL

Change the "(\*)" to a user defined prefix in the customer exit program.

```

01 (*)-BAT-STOR-DTCL.
05 (*)-BSDTCL-SORT-GROUP PIC 9(03).
05 (*)-BSDTCL-SORT-GROUP-FLG PIC X(01).
05 (*)-BSDTCL-FOLDER-ID PIC X(26).
05 (*)-BSDTCL-FOLDER-ID-FLG PIC X(01).
05 (*)-BSDTCL-FOLDER-TOKEN PIC X(26).
05 (*)-BSDTCL-FOLDER-TOKEN-FLG PIC X(01).
05 (*)-BSDTCL-CODE-PAGE-ID PIC 9(02) COMP.
05 (*)-BSDTCL-CODE-PAGE-ID-FLG PIC X(01).
05 (*)-BSDTCL-OSYSID PIC X(04).
05 (*)-BSDTCL-OSYSID-FLG PIC X(01).
05 (*)-BSDTCL-OVRD-IDX-FLG PIC X(01).
05 (*)-BSDTCL-OVRD-IDX-PASS-FLG PIC X(01).
05 (*)-BSDTCL-SEC-CLASS PIC X(02).
05 (*)-BSDTCL-SEC-CLASS-FLG PIC X(01).
05 (*)-BSDTCL-PAPER-RET PIC X(01).
05 (*)-BSDTCL-PAPER-RET-FLG PIC X(01).
05 (*)-BSDTCL-COLL-NAME PIC X(44).
05 (*)-BSDTCL-COLL-NAME-FLG PIC X(01).
05 (*)-BSDTCL-STOR-CLASS PIC X(08).
05 (*)-BSDTCL-STOR-CLASS-FLG PIC X(01).
05 (*)-BSDTCL-MGMT-CLASS PIC X(08).
05 (*)-BSDTCL-MGMT-CLASS-FLG PIC X(01).
05 (*)-BSDTCL-RETN-PER PIC 9(04) COMP.
05 (*)-BSDTCL-RETN-PER-FLG PIC X(01).
05 (*)-BSDTCL-DATE-REC PIC X(10).
05 (*)-BSDTCL-DATE-REC-FLG PIC X(01).
05 (*)-BSDTCL-DATE-FIL PIC X(10).
05 (*)-BSDTCL-DATE-FIL-FLG PIC X(01).
05 (*)-BSDTCL-USER-DATE PIC X(10).
05 (*)-BSDTCL-USER-DATE-FLG PIC X(01).
05 (*)-BSDTCL-ROUTE-FLG PIC X(01).
05 (*)-BSDTCL-ROUTE-PASS-FLG PIC X(01).
05 (*)-BSDTCL-OVRD-ROUTE PIC X(01).
05 (*)-BSDTCL-OVRD-ROUTE-FLG PIC X(01).
05 (*)-BSDTCL-USER-PARM1 PIC X(04).
05 (*)-BSDTCL-USER-PARM1-FLG PIC X(01).
05 (*)-BSDTCL-USER-PARM2 PIC 9(04) COMP.
05 (*)-BSDTCL-USER-PARM2-FLG PIC X(01).
05 (*)-BSDTCL-RLOB PIC X(06).
05 (*)-BSDTCL-RLOB-FLG PIC X(01).

```

05 (\*)-BSDTCL-TRAN-TYPE PIC X(06).  
05 (\*)-BSDTCL-TRAN-TYPE-FLG PIC X(01).  
05 (\*)-BSDTCL-OVRD-PRIOR-IND PIC X(01).  
05 (\*)-BSDTCL-OVRD-PRIOR-INDFLG PIC X(01).  
05 (\*)-BSDTCL-PRIORITY PIC 9(02) COMP.  
05 (\*)-BSDTCL-PRIORITY-FLG PIC X(01).  
05 (\*)-BSDTCL-ASSGN-USER-ID PIC X(08).  
05 (\*)-BSDTCL-ASSGN-USER-ID-FLG PIC X(01).  
05 (\*)-BSDTCL-HOLD-DATE PIC X(10).  
05 (\*)-BSDTCL-HOLD-DATE-FLG PIC X(01).  
05 (\*)-BSDTCL-HOLD-TIME PIC X(08).  
05 (\*)-BSDTCL-HOLD-TIME-FLG PIC X(01).  
05 (\*)-BSDTCL-UNIT-CODE PIC 9(04) COMP.  
05 (\*)-BSDTCL-UNIT-CODE-FLG PIC X(01).  
05 (\*)-BSDTCL-ROUTE-CODE PIC X(06).  
05 (\*)-BSDTCL-ROUTE-CODE-FLG PIC X(01).  
05 (\*)-BSDTCL-AGING-DATE PIC X(10).  
05 (\*)-BSDTCL-AGING-DATE-FLG PIC X(01).  
05 (\*)-BSDTCL-NUM-PAGES PIC 9(02) COMP.  
05 (\*)-BSDTCL-NUM-PAGES-FLG PIC X(01).

## Program Processing Logic

The VSR validation input file (INDATA1) contains the actual data to be processed by the Validation process (*AIBS002P*). The Validation process passes some of the fields in the file to the VSR Data Collection exit (*AIEX008P*) if both the CUSTEXIT and USEREXIT fields in the control file are set to "0". If a field is modified in the exit, the flag associated with that field should be set to 1. The validation process replaces the value supplied in the input file with the new value from the exit. The length fields, such as object description length, notes length, and user data length, are also replaced when the flag associated with that field is set to 1.

Set the return code in the exit to one of the following return codes. The VSR process takes an action based on the return code it receives.

- 00** Continue processing. No values are modified.
- 01** Continue processing. Use the values updated by the exit if the corresponding flags are set.
- 12** Continue processing. Reject the object.
- 16** Stop processing.
- 20** A DB2 deadlock or timeout (-911 or -913) was encountered, VSR should perform a restart.

**Note:** When customizing the exit if SQL code is added then you must test for DB2 deadlock and timeout conditions (-911 or -913 sqlcode return). If these conditions are encountered then the exit should send a return code of +20 to indicate to VSR that a retry should be attempted. Failure to test for this condition and return the proper return code will prevent the VSR process from controlling the commit scope of the work, and inconsistent results will be produced.

## Customer Data Exit: For FAF Tables

This section explains how to customize the Customer Data exit. This exit is called only when the value of USEREXIT is set to "0". The implementation of the Customer Data exit is similar to the implementation of the exit for the on-line system. The call to the Customer Data exit takes place in the update/registration process, after the document has been stored. The Customer Data exit:

Overrides the AIS+ application values for the folder, document, and workflow information, and provides user parameter values.

Determines whether a folder can be created dynamically. It can also specify the folder ID to be used.

The exit is called with an option code that is set to a value from 1 to 5. The option code determines what data is passed and whether the data is modified. Based on the option code, the Customer Data exit returns different data.

The following list describes the function for each option code:

### Option

### Code Description

1 The Customer Data exit returns the user parameter 1 and user parameter 2 fields that may be used to create the routing queue information.

5 The Customer Data exit determines whether the folder can be created dynamically. If the folder can be created, the exit specifies information about the folder, such as folder ID, folder type, description, security class, and secondary indexes.

Note: User parameter 1 and user parameter 2 are required fields to access the RLOB/Transaction Type table (AISRLTT) and Unit Code table (AISUNIT). You may define a Customer Data exit to provide these values. If your Customer Data exit does not provide these values, AIS+ uses spaces for user parameter 1 and zero for user parameter 2 to access these tables.

### Call Statement

The call statement used for the Customer Data exit is as follows

```
CALL - ( VSR Customer Data exit ) USING
      VSR Customer Data
ON EXCEPTION
      exception process
END-CALL.
```



### Parameter List Structure

The table below defines the fields passed to the Customer Data exit for different values of the option code. It provides the format of each field and specifies if the field is used as input or output or both.

If the exit modifies a field that has an associated flag, the exit must set the corresponding flag to 1. If the exit leaves a field with an associated flag unchanged, the exit must set the corresponding flag to 0.

**Note:** The COBOL variable name for each field is based on the usage of the copy member AILCSDTA supplied with AIS+. There is a prefix for each COBOL variable name in the copy member that is user definable.

**Note:** The format of the *AIS+ Customer Data Exit* copybook has been modified in AIS+ EE 1.0 to give users more flexibility in accessing data from subsequent exit calls. Therefore, you must be sure to compile your customized exit with the EE version of the copybook.

Parameter Descriptions for the Customer Data Exit				
Parameter Name	Option Code	Format	I/O	Description
Aging date	N/A	Char (10)	N/A	The aging date is used to determine the priority of a document.
Aging date flag	N/A	Char (1)	N/A	The aging date flag can have two values: 0 - Aging date field unchanged 1 - Aging date field changed.
Application ID	1,5	Bin (2)	I	The application ID code identifies the application from which the exit is being called.
Assigned user ID	N/A	Char (8)	N/A	The assigned user ID field contains the ID of the employee assigned to work on the document.
Assigned user ID flag	N/A	Char (1)	N/A	The assigned user ID flag can have two values: 0 - Assigned user ID field unchanged 1 - Assigned user ID field changed
Awake document	N/A	Char (1)	N/A	The awake document field indicates whether the suspended documents in the folder should be made available for processing. The awake document indicators are: 0 - Suspended documents in the folder should not be made available for processing 1 - Suspended documents in the folder should be made available for processing.
Create date	N/A	Char (10)	N/A	The create date field contains the creation date of the folder.

<b>Parameter Descriptions for the Customer Data Exit</b>				
<b>Parameter Name</b>	<b>Option Code</b>	<b>Format</b>	<b>I/O</b>	<b>Description</b>
Date format	1,5	Char (1)	I	The date format field contains the format used in the language specified for the application. The values are as follows: 1 - mm/dd/yyyy 2 - dd/mm/yyyy 3 - dd.mm.yyyy 4 - yyyy-mm-dd 5 - dd-mm-yyyy 6 - dd mm yyyy. The value of the date format field is input in the control file CNTRLDAT.
Document date filed	N/A	Char (10)	N/A	The document date filed field contains the date the document was filed in AIS+ and the Folder Application Facility.
Document date filed flag	N/A	Char (1)	N/A	The document date filed flag can have two values: 0 - Document date filed field unchanged 1 - Document date filed field changed
Document date received	N/A	Char (10)	N/A	The document date received field contains the date when a document was added to AIS+ and the Folder Application Facility.
Document date received flag	N/A	Char (1)	N/A	The document date received flag can have two values: 0 - Document date received field unchanged 1 - Document date received field changed
Document description	5	Char (60)	I	The document description field contains the description given to a document.
Document description flag	N/A	Char (1)	N/A	The document description flag can have two values: 0 - Document description field unchanged 1 - Document description field changed
Document description length	5	Bin (2)	I	The document description length field contains the length of the document description.
Document security class	N/A	Char (2)	N/A	The document security class field contains the security class assigned to the document.
Document security class flag	N/A	Char (1)	N/A	The document security class flag can have two values: 0 - Document security class field unchanged 1 - Document security class field changed

Parameter Descriptions for the Customer Data Exit				
Parameter Name	Option Code	Format	I/O	Description
Expiration date	N/A	Char (10)	N/A	The expiration date field contains the last date by which the document must be processed.
Expiration date flag	N/A	Char (1)	N/A	The expiration date flag can have two values: 0 - Expiration date field unchanged 1 - Expiration date field changed.
File tab	N/A	Char (16)	N/A	The file tab contains the name of the file tab in the folder under which the document is stored.
File tab flag	N/A	Char (1)	N/A	The file tab flag can have two values: 0 - File tab field unchanged 1 - File tab field changed.
Folder description	5	Char (60)	I/O	The folder description length field contains the descriptive text given to the folder.
Folder description length	5	Bin (2)	I/O	The folder description length field contains the actual length of the folder description.
Folder ID	1,5	Char (26)	I/O	The folder ID field contains the unique identifier of the folder in the Folder Application Facility application.
Folder ID flag	1,5	Char (1)	O	The folder ID flag can have two values: 0 - Folder ID field unchanged 1 - Folder IF field changed.
Folder ID length	1,5	Bin (2)	I/O	The folder ID length field contains the actual length of the folder ID.
Folder secondary index 1	5	Char (40)	I/O	The folder secondary index 1 field contains the value that groups folders within an application.
Folder secondary index 1 flag	5	Char (1)	O	The folder secondary index 1 flag can have two values: 0 - Folder secondary index 1 field unchanged 1 - Folder secondary index 1 field changed. <b>Note:</b> If the secondary index is not defined for the application, the Folder Application Facility ignores the changed folder secondary index value.
Folder secondary index 1 length	5	Bin (2)	I/O	The folder secondary index 1 length field contains the actual length of folder secondary index 1.

<b>Parameter Descriptions for the Customer Data Exit</b>				
<b>Parameter Name</b>	<b>Option Code</b>	<b>Format</b>	<b>I/O</b>	<b>Description</b>
Folder secondary index 2	5	Char (40)	I/O	The folder secondary index 2 field contains the value that groups folders within an application.
Folder secondary index 2 flag	5	Char (1)	O	The folder secondary index 2 flag can have two values: 0 - Folder secondary index 2 field unchanged 1 - Folder secondary index 2 field changed.
Folder secondary index 2 length	5	Bin (2)	I/O	The folder secondary index 2 field length contains the actual length of folder secondary index 2.
Folder secondary index 3	5	Char (40)	I/O	The folder secondary index 3 field contains the value that groups folders within an application.
Folder secondary index 3 flag	5	Char (1)	O	The folder secondary index 3 flag can have two values: 0 - Folder secondary index 3 field unchanged 1 - Folder secondary index 3 field changed.
Folder secondary index 3 length	5	Bin (2)	I/O	The folder secondary index 3 length field contains the actual length of folder secondary index 3.
Folder security class	5	Char (2)	I/O	The folder security class field contains the security class assigned to the folder.
Folder security class flag	5	Char (1)	O	The folder security class flag can have two values: 0 - Folder security class field unchanged 1 - Folder security class field changed.
Folder type	5	Char (8)	I/O	The folder type field contains the value used for classifying folders.
Folder type flag	5	Char (1)	O	The folder type flag can have two values: 0 - Folder type field unchanged 1 - Folder type field changed.
Form number	N/A	Char (16)	N/A	The form number field contains a code that identifies the type of document.
Form number flag	N/A	Char (1)	N/A	The form number flag can have two values: 0 - Form number field unchanged 1 - Form number field changed.
Function code	1,5	Char (2)	I	The function code field specifies the Folder Application Facility function that calls the exit.

Parameter Descriptions for the Customer Data Exit				
Parameter Name	Option Code	Format	I/O	Description
Hold date	N/A	Char (10)	N/A	The hold date field contains the date until which a routed document is on hold.
Hold date flag	N/A	Char (1)	N/A	The hold date flag can have two values: 0 - Hold date field unchanged 1 - Hold date field changed.
Hold time	N/A	Char (8)	N/A	The hold time field contains the time on the hold date until which the document is on hold.
Hold time flag	N/A	Char (1)	N/A	The hold time flag can have two values: 0 - Hold time field unchanged 1 - Hold time field changed.
IODM ID	1,5	Char (4)	I	The IODM ID field contains the Object Distribution Manager system ID.
Language ID	1,5	Char (3)	I	The language ID field contains the identifier of the language used to communicate with the user. <b>AIS+ does not use this field.</b>
Line 3 data	1,5	Char (78)	I	The line 3 data field contains 72 bytes of the Notes field starting in position 35.
Message code	1,5	Char (8)	O	The message code field contains the message code generated by the user exit when the return code is 12.
Option code	1,5	Char (1)	I	The option code is set to a value from 1 to 5. The Folder Application Facility calling program tells the Customer Data exit what to do based on the value of the option code.
Override priority indicator	N/A	Char (1)	N/A	The override priority indicator field contains the value used to set the priority of a routed document. The values are: 0 Normal 1 - Low 2 - Medium 3 - High.
Override priority indicator flag	N/A	Char (1)	N/A	The override priority indicator flag can have two values: 0 - Override priority indicator field unchanged 1 - Override priority indicator field changed.
Paper kept	N/A	Char (1)	N/A	The paper kept field indicates whether the physical copy of the document should be kept after the document is stored in the Folder Application Facility.

<b>Parameter Descriptions for the Customer Data Exit</b>				
<b>Parameter Name</b>	<b>Option Code</b>	<b>Format</b>	<b>I/O</b>	<b>Description</b>
Paper kept flag	N/A	Char (1)	N/A	The paper kept flag can have two values: 0 - Paper kept field unchanged 1 - Paper kept field changed.
Return code	1,5	Num (2)	O	The return code must be set to one of the following values that controls the subsequent processing: 00 - Continue processing 01 - Continue processing. Use the values updated by the exit. 12 - Stop processing this function and display a message.
RLOB	N/A	Char (6)	N/A	The RLOB field specifies the routing line-of-business used to generate the routing destination for the document.
RLOB flag	N/A	Char (1)	N/A	The RLOB flag can have two values: 0 - RLOB field unchanged 1 - RLOB field changed.
Route code	N/A	Char (6)	N/A	The route code field contains the route code value which, along with the unit code value, determines which routing queue the document is placed in.
Route code flag	N/A	Char (1)	N/A	The route code flag can have two values: 0 - Route code field unchanged 1 - Route code field changed.
Routing decision	N/A	Char (1)	N/A	The routing decision field indicates whether to add the document or add and route the document. The routing decision indicators are: 0 - Add the document to the folder 1 - Add the document to the folder
Routing decision flag	N/A	Char (1)	N/A	The routing decision flag field can have two values: 0 - Routing decision field unchanged 1 - Routing decision field changed.

<b>Parameter Descriptions for the Customer Data Exit</b>				
<b>Parameter Name</b>	<b>Option Code</b>	<b>Format</b>	<b>I/O</b>	<b>Description</b>
Supervisory authority	N/A	Char (1)	N/A	The supervisory authority field indicates whether the user can perform supervisory functions. The supervisory authority indicators are: 0 - Cannot perform supervisory functions 1 - Can perform supervisory functions.
Time format	1,5	Char (1)	I	The time format field contains the format used in the language specified for the application. The values are as follows: 1 - 12-hour format (hh:mm xx) 2 - 24-hour format (hh:mm ) For 12-hour time, a number from 01 to 12 specifies the hour (hh), 00 to 59 specifies the minutes (mm), and AM specifies a.m. or PM specifies p.m. The colon (:) and space ( ) are required. For 24-hour time, a number from 00 to 23 specifies the hour (hh) and 00 to 59 specifies the minutes (mm). The colon (:) and three spaces ( ) are required. For example, to specify 10:30 in the evening, the time parameter value is: 10:30 PM for 12-hour time 22:30 for 24-hour time. The value of the time format field is input in the control file CNTRLDAT.
Transaction type	N/A	Char (6)	N/A	The transaction type field contains a classification of the document indicating the type of work that must be performed on the document.
Transaction type flag	N/A	Char (1)	N/A	The transaction type flag can have two values: 0 - Transaction type field unchanged 1 - Transaction type field changed.
Unit code	N/A	Bin (4)	N/A	The unit code field contains the unit code value which, along with the route code value, determines which queue the document is routed to for processing.
Unit code flag	N/A	Char (1)	N/A	The user code flag can have two values: 0 - User code field unchanged 1 - User code field changed.

<b>Parameter Descriptions for the Customer Data Exit</b>				
<b>Parameter Name</b>	<b>Option Code</b>	<b>Format</b>	<b>I/O</b>	<b>Description</b>
User date	N/A	Char (10)	N/A	The user date field contains a user-defined date associated with a document.
User date flag	N/A	Char (1)	N/A	The user date flag can have two values: 0 - User date field unchanged 1 - User date field changed.
User exit area	1,5	Char (20)	I/O	The user exit area field contains the data passed from one user exit to another.
User exit data	1,5	Char (20)	I	The user exit data field contains the data that is captured on the Folder Functions panel and the Documents Functions panel. The captured data is passed to the user exits.
User ID	N/A	Char (8)	N/A	The user ID field contains the ID that identifies the user to the Folder Application Facility function for which the exit is being called.
User parameter 1	1	Char (4)	I/O	The user parameter 1 field contains a user-defined parameter.
User parameter 2	1	Char (4)	I/O	The user parameter 2 field contains a user-defined parameter.
User security class	N/A	Char (2)	N/A	The user security class field contains the security class value assigned to the user.



## Copy member AILCSDTA

Change the "\*" to a user defined prefix in the customer exit program.

```

05 (*)-CSDT-PARMS.
10 (*)-CSDT-CODE PIC X(01).
10 (*)-CSDT-FUNCTIONCD PIC X(02).
10 (*)-CSDT-USERID PIC X(08).
10 (*)-CSDT-APPLIDCD PIC S9(04) COMP.
10 (*)-CSDT-LANGID PIC X(03).
10 (*)-CSDT-SUPAUTH PIC X(01).
10 (*)-CSDT-USERSECCL PIC X(02).
10 (*)-CSDT-IODMID PIC X(04).
10 (*)-CSDT-USER-EXIT-DATA PIC X(20).
10 (*)-CSDT-EXIT-AREA PIC X(20).
10 (*)-CSDT-DATE-FORMAT PIC X(01).
10 (*)-CSDT-TIME-FORMAT PIC X(01).
10 (*)-CSDT-RETURN-CODE PIC 9(02).
10 (*)-CSDT-MESSAGE-CODE PIC X(08).
10 (*)-CSDT-LINE-3-DATA PIC X(78).
10 (*)-CSDT-USER-PARM-1 PIC X(04).
10 (*)-CSDT-USER-PARM-1-FLAG PIC X(01).
88 (*)-CSDT-YES-USER-PARM-1 VALUE '1'.
88 (*)-CSDT-NO-USER-PARM-1 VALUE '0'.
10 (*)-CSDT-USER-PARM-2 PIC S9(09) COMP.
10 (*)-CSDT-USER-PARM-2-FLAG PIC X(01).
88 (*)-CSDT-YES-USER-PARM-2 VALUE '1'.
88 (*)-CSDT-NO-USER-PARM-2 VALUE '0'.
10 (*)-CSDT-FOLDID-LEN PIC S9(04) COMP.
10 (*)-CSDT-FOLDID PIC X(26).
10 (*)-CSDT-FOLDID-FLAG PIC X(01).
88 (*)-CSDT-FOLDID-PRESENT VALUE '1'.
88 (*)-CSDT-NO-FOLDID VALUE '0'.
10 (*)-CSDT-FORMNO PIC X(16).
10 (*)-CSDT-FORMNO-FLAG PIC X(01).
88 (*)-CSDT-YES-FORMNO VALUE '1'.
88 (*)-CSDT-NO-FORMNO VALUE '0'.
10 (*)-CSDT-CODE-PARMS PIC X(270).
10 (*)-CSDT-CODEL-PARMS REDEFINES (*)-CSDT-CODE-PARMS.
15 FILLER PIC X(270).
10 (*)-CSDT-CODE2-PARMS REDEFINES (*)-CSDT-CODE-PARMS.
15 (*)-CSDT-C2-FOLDTYPE PIC X(08).
15 (*)-CSDT-C2-FOLDTYPE-FLAG PIC X(01).
88 (*)-CSDT-YES-C2-FOLDTYPE VALUE '1'.
88 (*)-CSDT-NO-C2-FOLDTYPE VALUE '0'.
15 (*)-CSDT-C2-FOLDSECCL PIC X(02).
15 (*)-CSDT-C2-FOLDSECCL-FLAG PIC X(01).
88 (*)-CSDT-YES-C2-FOLDSECCL VALUE '1'.
88 (*)-CSDT-NO-C2-FOLDSECCL VALUE '0'.
15 (*)-CSDT-C2-FOLDDESC-LEN PIC S9(04) COMP.
15 (*)-CSDT-C2-FOLDDESC PIC X(60).
15 (*)-CSDT-C2-FOLDDESC-FLAG PIC X(01).
88 (*)-CSDT-YES-C2-FOLDDESC VALUE '1'.
88 (*)-CSDT-NO-C2-FOLDDESC VALUE '0'.
15 (*)-CSDT-C2-FOLD-SECINDX1-LEN PIC S9(04) COMP.
15 (*)-CSDT-C2-FOLD-SECINDX1 PIC X(40).
15 (*)-CSDT-C2-FOLD-SECINDX1-FLAG PIC X(01).
88 (*)-CSDT-YES-C2-FOLD-SECINDX1 VALUE '1'.
88 (*)-CSDT-NO-C2-FOLD-SECINDX1 VALUE '0'.
15 (*)-CSDT-C2-FOLD-SECINDX2-LEN PIC S9(04) COMP.
15 (*)-CSDT-C2-FOLD-SECINDX2 PIC X(40).
15 (*)-CSDT-C2-FOLD-SECINDX2-FLAG PIC X(01).
88 (*)-CSDT-YES-C2-FOLD-SECINDX2 VALUE '1'.
88 (*)-CSDT-NO-C2-FOLD-SECINDX2 VALUE '0'.
15 (*)-CSDT-C2-FOLD-SECINDX3-LEN PIC S9(04) COMP.

```

```
15 (*)-CSDT-C2-FOLD-SECINDX3 PIC X(40).
15 (*)-CSDT-C2-FOLD-SECINDX3-FLAG PIC X(01).
88 (*)-CSDT-YES-C2-FOLD-SECINDX3 VALUE '1'.
88 (*)-CSDT-NO-C2-FOLD-SECINDX3 VALUE '0'.
15 (*)-CSDT-C2-CRTEDEDATE PIC X(10).
15 (*)-CSDT-C2-CRTEDEDATE-FLAG PIC X(01).
88 (*)-CSDT-YES-C2-CRTEDEDATE VALUE '1'.
88 (*)-CSDT-NO-C2-CRTEDEDATE VALUE '0'.
15 FILLER PIC X(55).
10 (*)-CSDT-CODE3-PARMS REDEFINES
(*)-CSDT-CODE-PARMS.
15 (*)-CSDT-C3-FORMNO PIC X(16).
15 (*)-CSDT-C3-FORMNO-FLAG PIC X(01).
88 (*)-CSDT-YES-C3-FORMNO VALUE '1'.
88 (*)-CSDT-NO-C3-FORMNO VALUE '0'.
15 (*)-CSDT-C3-FILETAB PIC X(16).
15 (*)-CSDT-C3-FILETAB-FLAG PIC X(01).
88 (*)-CSDT-YES-C3-FILETAB VALUE '1'.
88 (*)-CSDT-NO-C3-FILETAB VALUE '0'.
15 (*)-CSDT-C3-DOC-SECCL PIC X(02).
15 (*)-CSDT-C3-DOC-SECCL-FLAG PIC X(01).
88 (*)-CSDT-YES-C3-DOC-SECCL VALUE '1'.
88 (*)-CSDT-NO-C3-DOC-SECCL VALUE '0'.
15 (*)-CSDT-C3-DOC-RECVDATE PIC X(10).
15 (*)-CSDT-C3-DOC-RECVDATE-FLAG PIC X(01).
88 (*)-CSDT-YES-C3-DOC-RECVDATE VALUE '1'.
88 (*)-CSDT-NO-C3-DOC-RECVDATE VALUE '0'.
15 (*)-CSDT-C3-DOC-USERDATE PIC X(10).
15 (*)-CSDT-C3-DOC-USERDATE-FLAG PIC X(01).
88 (*)-CSDT-YES-C3-DOC-USERDATE VALUE '1'.
88 (*)-CSDT-NO-C3-DOC-USERDATE VALUE '0'.
15 (*)-CSDT-C3-DOC-FILEDATE PIC X(10).
15 (*)-CSDT-C3-DOC-FILEDATE-FLAG PIC X(01).
88 (*)-CSDT-YES-C3-DOC-FILEDATE VALUE '1'.
88 (*)-CSDT-NO-C3-DOC-FILEDATE VALUE '0'.
15 (*)-CSDT-C3-DOC-DESC-LEN PIC S9(04) COMP.
15 (*)-CSDT-C3-DOC-DESC PIC X(60).
15 (*)-CSDT-C3-DOC-DESC-FLAG PIC X(01).
88 (*)-CSDT-YES-C3-DOC-DESC VALUE '1'.
88 (*)-CSDT-NO-C3-DOC-DESC VALUE '0'.
15 (*)-CSDT-C3-PAPER-KEPT PIC X(01).
15 (*)-CSDT-C3-PAPER-KEPT-FLAG PIC X(01).
88 (*)-CSDT-YES-C3-PAPER-KEPT VALUE '1'.
88 (*)-CSDT-NO-C3-PAPER-KEPT VALUE '0'.
15 FILLER PIC X(135).
10 (*)-CSDT-CODE4-PARMS REDEFINES (*)-CSDT-CODE-PARMS.
15 (*)-CSDT-C4-FORMNO PIC X(16).
15 (*)-CSDT-C4-FORMNO-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-FORMNO VALUE '1'.
88 (*)-CSDT-NO-C4-FORMNO VALUE '0'.
15 (*)-CSDT-C4-FILETAB PIC X(16).
15 (*)-CSDT-C4-FILETAB-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-FILETAB VALUE '1'.
88 (*)-CSDT-NO-C4-FILETAB VALUE '0'.
15 (*)-CSDT-C4-DOC-SECCL PIC X(02).
15 (*)-CSDT-C4-DOC-SECCL-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-DOC-SECCL VALUE '1'.
88 (*)-CSDT-NO-C4-DOC-SECCL VALUE '0'.
15 (*)-CSDT-C4-RECEIVE-DATE PIC X(10).
15 (*)-CSDT-C4-RECEIVE-DATE-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-RECEIVE-DATE VALUE '1'.
88 (*)-CSDT-NO-C4-RECEIVE-DATE VALUE '0'.
15 (*)-CSDT-C4-DOC-USER-DATE PIC X(10).
15 (*)-CSDT-C4-DOC-USER-DATE-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-DOC-USER-DATE VALUE '1'.
```

```
88 (*)-CSDT-NO-C4-DOC-USER-DATE VALUE '0'.
15 (*)-CSDT-C4-DOC-DESC-LEN PIC S9(04) COMP.
15 (*)-CSDT-C4-DOC-DESC PIC X(60).
15 (*)-CSDT-C4-DOC-DESC-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-DOC-DESC VALUE '1'.
88 (*)-CSDT-NO-C4-DOC-DESC VALUE '0'.
15 (*)-CSDT-C4-PAPER-KEPT PIC X(01).
15 (*)-CSDT-C4-PAPER-KEPT-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-PAPER-KEPT VALUE '1'.
88 (*)-CSDT-NO-C4-PAPER-KEPT VALUE '0'.
15 (*)-CSDT-C4-RLOB PIC X(06).
15 (*)-CSDT-C4-RLOB-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-RLOB VALUE '1'.
88 (*)-CSDT-NO-C4-RLOB VALUE '0'.
15 (*)-CSDT-C4-TRAN-TYPE PIC X(06).
15 (*)-CSDT-C4-TRAN-TYPE-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-TRAN-TYPE VALUE '1'.
88 (*)-CSDT-NO-C4-TRAN-TYPE VALUE '0'.
15 (*)-CSDT-C4-UNIT-CODE PIC S9(09) COMP.
15 (*)-CSDT-C4-UNIT-CODE-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-UNIT-CODE VALUE '1'.
88 (*)-CSDT-NO-C4-UNIT-CODE VALUE '0'.
15 (*)-CSDT-C4-RCODE PIC X(06).
15 (*)-CSDT-C4-RCODE-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-RCODE VALUE '1'.
88 (*)-CSDT-NO-C4-RCODE VALUE '0'.
15 (*)-CSDT-C4-EXP-DATE PIC X(10).
15 (*)-CSDT-C4-EXP-DATE-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-EXP-DATE VALUE '1'.
88 (*)-CSDT-NO-C4-EXP-DATE VALUE '0'.
15 (*)-CSDT-C4-ASSIGN-USER PIC X(08).
15 (*)-CSDT-C4-ASSIGN-USER-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-ASSIGN-USER VALUE '1'.
88 (*)-CSDT-NO-C4-ASSIGN-USER VALUE '0'.
15 (*)-CSDT-C4-AGE-DATE PIC X(10).
15 (*)-CSDT-C4-AGE-DATE-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-AGE-DATE VALUE '1'.
88 (*)-CSDT-NO-C4-AGE-DATE VALUE '0'.
15 (*)-CSDT-C4-HOLD-DATE PIC X(10).
15 (*)-CSDT-C4-HOLD-DATE-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-HOLD-DATE VALUE '1'.
88 (*)-CSDT-NO-C4-HOLD-DATE VALUE '0'.
15 (*)-CSDT-C4-HOLD-TIME PIC X(08).
15 (*)-CSDT-C4-HOLD-TIME-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-HOLD-TIME VALUE '1'.
88 (*)-CSDT-NO-C4-HOLD-TIME VALUE '0'.
15 (*)-CSDT-C4-OVR-PRTY-IND PIC X(01).
15 (*)-CSDT-C4-OVR-PRTY-IND-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-OVR-PRTY-IND VALUE '1'.
88 (*)-CSDT-NO-C4-OVR-PRTY-IND VALUE '0'.
15 (*)-CSDT-C4-ROUT-DECISION PIC X(01).
15 (*)-CSDT-C4-ROUT-DECISION-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-ROUT-DECISION VALUE '1'.
88 (*)-CSDT-NO-C4-ROUT-DECISION VALUE '0'.
15 (*)-CSDT-C4-AWAKEDOC PIC X(01).
15 (*)-CSDT-C4-AWAKEDOC-FLAG PIC X(01).
88 (*)-CSDT-YES-C4-AWAKEDOC VALUE '1'.
88 (*)-CSDT-NO-C4-AWAKEDOC VALUE '0'.
15 FILLER PIC X(63).
10 (*)-CSDT-CODE5-PARMS REDEFINES (*)-CSDT-CODE-PARMS.
15 (*)-CSDT-C5-FOLDTYPE PIC X(08).
15 (*)-CSDT-C5-FOLDTYPE-FLAG PIC X(01).
88 (*)-CSDT-YES-C5-FOLDTYPE VALUE '1'.
88 (*)-CSDT-NO-C5-FOLDTYPE VALUE '0'.
15 (*)-CSDT-C5-FOLDSECCL PIC X(02).
```

```
15 (*)-CSDT-C5-FOLDSECCL-FLAG PIC X(01).
88 (*)-CSDT-YES-C5-FOLDSECCL VALUE '1'.
88 (*)-CSDT-NO-C5-FOLDSECCL VALUE '0'.
15 (*)-CSDT-C5-FOLDDESC-LEN PIC S9(04) COMP.
15 (*)-CSDT-C5-FOLDDESC PIC X(60).
15 (*)-CSDT-C5-FOLDDESC-FLAG PIC X(01).
88 (*)-CSDT-YES-C5-FOLDDESC VALUE '1'.
88 (*)-CSDT-NO-C5-FOLDDESC VALUE '0'.
15 (*)-CSDT-C5-FOLD-SECINDX1-LEN PIC S9(04) COMP.
15 (*)-CSDT-C5-FOLD-SECINDX1 PIC X(40).
15 (*)-CSDT-C5-FOLD-SECINDX1-FLAG PIC X(01).
88 (*)-CSDT-YES-C5-FOLD-SECINDX1 VALUE '1'.
88 (*)-CSDT-NO-C5-FOLD-SECINDX1 VALUE '0'.
15 (*)-CSDT-C5-FOLD-SECINDX2-LEN PIC S9(04) COMP.
15 (*)-CSDT-C5-FOLD-SECINDX2 PIC X(40).
15 (*)-CSDT-C5-FOLD-SECINDX2-FLAG PIC X(01).
88 (*)-CSDT-YES-C5-FOLD-SECINDX2 VALUE '1'.
88 (*)-CSDT-NO-C5-FOLD-SECINDX2 VALUE '0'.
15 (*)-CSDT-C5-FOLD-SECINDX3-LEN PIC S9(04) COMP.
15 (*)-CSDT-C5-FOLD-SECINDX3 PIC X(40).
15 (*)-CSDT-C5-FOLD-SECINDX3-FLAG PIC X(01).
88 (*)-CSDT-YES-C5-FOLD-SECINDX3 VALUE '1'.
88 (*)-CSDT-NO-C5-FOLD-SECINDX3 VALUE '0'.
15 (*)-CSDT-C5-CRTEDATE PIC X(10).
15 (*)-CSDT-C5-CRTEDATE-FLAG PIC X(01).
88 (*)-CSDT-YES-C5-CRTEDATE VALUE '1'.
88 (*)-CSDT-NO-C5-CRTEDATE VALUE '0'.
15 (*)-CSDT-C5-CREATE-FOLDER-FLAG PIC X(01).
88 (*)-CSDT-YES-C5-CREATE-FOLDER VALUE '1'.
88 (*)-CSDT-NO-C5-CREATE-FOLDER VALUE '0'.
15 FILLER PIC X(54).
10 (*)-CSDT-CODE-5-PARMS2.
15 (*)-CSDT-C5-OBJECT-DESCRIPTION PIC X(253).
15 (*)-CSDT-C5-OBJECT-DESC-FLAG PIC X(01).
88 (*)-CSDT-YES-C5-OBJDESC
VALUE '1'.
88 (*)-CSDT-NO-C5-OBJDESC
VALUE '0'.
15 (*)-CSDT-C5-OBJECT-LENGTH PIC S9(4) COMP.
```

## Program Processing Logic

When your enterprise installs AIS+ it defines the folder ID, folder type, folder description, security class, RLOB, transaction type, and secondary indexes. As the VSR facility processes a specific record, it obtains information from the index area and the AIS+ tables. When the VSR facility passes this information to the Customer Data exit, you may change some of the information. If you modify a field, the flag associated with that field should be set to 1. The VSR facility utilizes this information in processing the record.

When processing is completed, Set the return code field to one of the following values:

**00** - Normal completion.

**01** - Continue processing. Use the values updated by the exit.

**16** - Stop processing and display a message. The message code that describes the problem is contained in the message code field.

**20** - A DB2 deadlock or timeout (-911 or -913) was encountered. VSR will perform a restart.

**Note:** When customizing the exit if SQL code is added then you must test for DB2 deadlock and timeout conditions (-911 or -913 sqlcode return). If these conditions are encountered then the exit should send a return code of +20 to indicate the VSR that a retry should be attempted. Failure to test for this condition and return the proper return code will prevent the VSR process from controlling the commit scope of the work.

## Routing Data Exit: For FAF Tables

The Routing Data exit can override the current routing information when a document is initially routed. The Routing Data exit is called in the update/registration process when a document is stored and before it is routed. This exit is called only when the value of USEREXIT is set to "0". This user exit is taken after the AIS+ facility has derived a unit code, route code, and base priority for a document but before the document is placed on the appropriate queue for processing. It is used to route a document to the appropriate work queue.

### Call Statement

The call statement used for the Routing Data exit is as follows:

```
CALL - ( VSR Routing Data exit ) USING
      VSR Routing Data
ON EXCEPTION
      exception process
END-CALL.
```

### Parameter List Structure

The table below defines the fields passed to the Routing Data exit. It provides the format of each field and specifies whether the field is used as input or output or both. If the exit modifies a field that has an associated flag, the exit must set the corresponding flag to 1. If the exit leaves a field with an associated flag unchanged, the exit must set the corresponding flag to 0.

**Note:** The COBOL variable name for each field is based on the usage of the copy member AILROUTE supplied with AIS+. There is a prefix for each COBOL variable name in the copy member that is user definable.

Parameter Descriptions for the Routing Data Exit			
Parameter Name	Format	I/O	Description
Aging date	Char (10)	I/O	The aging date is used to calculate the priority of a document.
Aging date flag	Char (1)	O	The aging date flag can have two values: 0 - Aging date field unchanged. 1 - Aging date field changed.
Application ID code	Bin (2)	I	The application ID code field identifies the application from which the exit is being called.
Assigned user ID	Char (8)	I/O	The assigned user ID field contains the user ID of the employee assigned to process the document.
Assigned User ID Flag	Char (1)	O	The assigned user ID flag can have two values: 0 - User ID field unchanged. 1 - User ID field changed.

<b>Parameter Descriptions for the Routing Data Exit</b>			
<b>Parameter Name</b>	<b>Format</b>	<b>I/O</b>	<b>Description</b>
Date format	Char (1)	I	The date format field contains the format used in the language specified for the application. The values are as follows: 1 - mm/dd/yyyy 2 - dd/mm/yyyy 3 - dd.mm.yyyy 4 - yyyy-mm-dd 5 - dd-mm-yyyy 6 - dd mm yyyy The value of the date format field is input in the control file CNTRLDAT.
Expiration date	Char (10)	I/O	The expiration date field contains the last date that the document must be processed.
Expiration date flag	Char (1)	O	The expiration date flag can have two values: 0 - Date field unchanged. 1 - Date field changed.
Folder ID	Char (26)	I	The folder ID field contains the unique identifier of the folder in the AIS+ System
Folder ID Length	Bin (2)	I	The folder ID length contains the actual length of the folder ID.
Function code	Char (2)	I	The function code field specifies the function that calls the exit.
Hold date	Char (10)	I/O	The hold date field contains a date until which a routed document is on hold.
Hold date flag	Char (1)	O	The hold date flag can have two values: 0 - Hold Date field unchanged. 1 - Hold Date field changed.
Hold Time	Char (8)	I/O	The hold time field contains the time of the hold date until which a routed document is on hold.
Hold Time flag	Char (1)	O	The hold time flag can have two values: 0 - Hold Time field unchanged. 1 - Hold Time field changed.
Language ID	Char (3)	I	The language ID field is not used in current versions of AIS+.

<b>Parameter Descriptions for the Routing Data Exit</b>			
<b>Parameter Name</b>	<b>Format</b>	<b>I/O</b>	<b>Description</b>
Message code	Char (8)	O	The message code field contains the message code generated by the user exit when the return code is 12.
Number of documents	Num (1)	I	The number of documents field shows the number of documents that the exit is going to provide routing information for. See the document-specific details section.
Override priority indicator	Char (1)	O	The override priority indicator flag can have two values: 0 - Override priority indicator field unchanged. 1 - Override priority indicator field changed.
Priority	Bin (2)	I/O	The priority field contains a value from 0 to 999. Depending on the priority value, the document is selected for processing.
Priority Flag	Char (1)	O	The priority flag can have two values: 0 - Override priority unchanged. 1 - Override priority field changed.
Return code	Num (2)	O	The return code must be set to one of the following values that controls subsequent processing: 00 - Continue processing 01 - Continue processing. Use the values updated by the exit 12 - Stop processing this function and display a message.
RLOB	Char (6)	I/O	The RLOB field specifies the routing line-of-business used to generate the routing destination for a document.
RLOB flag	Char (1)	O	The RLOB flag can have two values: 0 - RLOB unchanged. 1 - RLOB changed.
Route Code	Char (6)	I/O	The route code field contains the route code value which along with the unit code value determines which routing queue the document is placed in.
Route Code flag	Char (1)	O	The route code flag can have two values: 0 - Route code unchanged. 1 - Route code changed.



<b>Parameter Descriptions for the Routing Data Exit</b>			
<b>Parameter Name</b>	<b>Format</b>	<b>I/O</b>	<b>Description</b>
Supervisory Authority	Char (1)	I	The supervisory authority field indicates whether the user can perform supervisory functions. The supervisory authority indicators are: 0 - Cannot perform supervisory functions. 1 - Can perform supervisory functions.
Time format	Char (1)	I	The time format field contains the format used in the language specified for the application. The values are as follows: 1 - 12-hour format (hh:mm xx) 2 - 24-hour format (hh:mm) For 12 hour time, a number from 01 to 12 specifies the hour (hh), 00 to 59 specifies the minutes (mm), and AM specifies a.m. or PM specifies p.m. The colon (:) and three spaces ( ) are required. For example, to specify 10:30 in the evening, the time parameter value is: 10:30 PM for 12-hour time 22:30 for 24-hour time. The value of the time format field is input in the control file CNTRLDAT.
Transaction type	Char (6)	I/O	The transaction type field contains a classification of the document indicating the type of work that must be performed on the document.
Transaction type flag	Char (1)		The transaction type flag can have two values: 0 - Transaction type unchanged. 1 - Transaction type changed.
Unit Code	Bin (4)	I/O	The unit code field contains the unit code value which, along with the route code value, determines which queue the document is routed to for processing.
Unit Code Flag	Char (1)	O	The Unit Code flag can have two values: 0 - Unit code unchanged. 1 - Unit code changed.
User exit area	Char (20)	I/O	The user exit area field contains the data passed from one user exit to the next.
User ID	Char (8)	I	The user ID field contains the ID that identifies the user to the AIS Plus function for which the exit is being called.

<b>Parameter Descriptions for the Routing Data Exit</b>			
<b>Parameter Name</b>	<b>Format</b>	<b>I/O</b>	<b>Description</b>
User Parameter 1	Char (4)	I/O	The user parameter 1 field contains a user-defined parameter.
User Parameter 1 flag	Char (1)	O	The User Parameter 1 flag can have two values: 0 - Parameter 1 unchanged. 1 - Parameter 1 changed.
User Parameter 2	Char (4)	I/O	The user parameter 2 field contains a user-defined parameter.
User Parameter 2 Flag	Char (1)	O	The User Parameter 2 flag can have two values: 0 - Parameter 2 unchanged. 1 - Parameter 2 changed.
User security class	Char (2)	I	The user security class field contains the security class assigned to the user.
Document-specific details(occurs as a group six times). VSR will always use the first occurs group only.			
Document date received	Char (10)	I	The document date received field contains the date when document was added to the AIS Plus system.
Document security class	Char (2)	I	The document security class field contains the security class assigned to the document.
File tab	Char (8)	I	The file tab field contains the name of the name of the file tab for which the document is stored.
Form number	Char (10)	I	The form number field contains a code that identifies the type of document.
Object Token	Char (30)	I	The object token field contains a unique identifier generated by the Folder Application Facility API. It is used when a document is added.
User date	Char (10)	I	The user date field contains a date assigned by the user.

## Copy member AILROUTE

Change the "\*" to a user defined prefix in the customer exit program.

```
05 (*)-RTE-PARMS.
10 (*)-RTE-FUNCTIONCD PIC X(02).
10 (*)-RTE-USERID PIC X(08).
10 (*)-RTE-APPLIDCD PIC S9(04) COMP.
10 (*)-RTE-LANGID PIC X(03).
10 (*)-RTE-SUPAUTH PIC X(01).
10 (*)-RTE-USERSECCL PIC X(02).
10 (*)-RTE-USER-EXIT-AREA PIC X(20).
10 (*)-RTE-FOLDID PIC X(26).
10 (*)-RTE-FOLDID-LEN PIC S9(04) COMP.
10 (*)-RTE-FOLDER-TOKEN PIC X(26).
10 (*)-RTE-NO-OF-OBJECTS PIC 9(01).
10 (*)-RTE-DOC-DETAILS-ARRAY OCCURS 6 TIMES.
15 (*)-RTE-DOC-FORMNO PIC X(10).
15 (*)-RTE-DOC-FILETAB PIC X(08).
15 (*)-RTE-DOC-OBJTKN PIC X(30).
15 (*)-RTE-DOC-SECCL PIC X(02) .
15 (*)-RTE-DOC-RECVDATE PIC X(10).
15 (*)-RTE-DOC-USER-DATE PIC X(10).
10 (*)-RTE-ROUTING-PARMS.
15 (*)-RTE-DOC-RLOB PIC X(06).
15 (*)-RTE-DOC-RLOB-FLAG PIC X(01).
88 (*)-C-DOC-RLOB VALUE '1'.
88 (*)-C-DOC-NO-RLOB VALUE '0'.
15 (*)-RTE-DOC-TRAN-TYPE PIC X(06).
15 (*)-RTE-DOC-TRAN-TYPE-FLAG PIC X(01).
88 (*)-C-DOC-TRAN-TYPE VALUE '1'.
88 (*)-C-DOC-NO-TRAN-TYPE VALUE '0'.
15 (*)-RTE-DOC-UNIT-CODE PIC S9(09) COMP.
15 (*)-RTE-DOC-UNIT-CODE-FLAG PIC X(01).
88 (*)-C-DOC-UNIT-CODE VALUE '1'.
88 (*)-C-DOC-NO-UNIT-CODE VALUE '0'.
15 (*)-RTE-DOC-ROUTE-CODE PIC X(06).
15 (*)-RTE-DOC-ROUTE-CODE-FLAG PIC X(01).
88 (*)-C-DOC-ROUTE-CODE VALUE '1'.
88 (*)-C-DOC-NO-ROUTE-CODE VALUE '0'.
15 (*)-RTE-DOC-AGE-DATE PIC X(10).
15 (*)-RTE-DOC-AGE-DATE-FLAG PIC X(01).
88 (*)-C-DOC-AGE-DATE VALUE '1'.
88 (*)-C-DOC-NO-AGE-DATE VALUE '0'.
15 (*)-RTE-DOC-HOLD-DATE PIC X(10).
15 (*)-RTE-DOC-HOLD-DATE-FLAG PIC X(01).
88 (*)-C-DOC-HOLD-DATE VALUE '1'.
88 (*)-C-DOC-NO-HOLD-DATE VALUE '0'.
15 (*)-RTE-DOC-HOLD-TIME PIC X(08).
15 (*)-RTE-DOC-HOLD-TIME-FLAG PIC X(01).
88 (*)-C-DOC-HOLD-TIME VALUE '1'.
88 (*)-C-DOC-NO-HOLD-TIME VALUE '0'.
15 (*)-RTE-DOC-ASSIGN-USER PIC X(08).
15 (*)-RTE-DOC-ASSIGN-USER-FLAG PIC X(01).
88 (*)-C-DOC-ASSIGN-USER VALUE '1'.
88 (*)-C-DOC-NO-ASSIGN-USER VALUE '0'.
15 (*)-RTE-DOC-EXP-DATE PIC X(10).
15 (*)-RTE-DOC-EXP-DATE-FLAG PIC X(01).
88 (*)-C-DOC-EXP-DATE VALUE '1'.
88 (*)-C-DOC-NO-EXP-DATE VALUE '0'.
15 (*)-RTE-DOC-PRIORITY PIC S9(04) COMP.
15 (*)-RTE-DOC-PRIORITY-FLAG PIC X(01).
88 (*)-C-DOC-PRIORITY VALUE '1'.
```

```
88 (*)-C-DOC-NO-PRIORITY VALUE '0'.
15 (*)-RTE-DOC-OVR-PRTY-IND PIC X(01).
15 (*)-RTE-DOC-OVR-PRTY-IND-FLAG PIC X(01).
88 (*)-C-DOC-OVR-PRTY-IND VALUE '1'.
88 (*)-C-DOC-NO-OVR-PRTY-IND VALUE '0'.
15 (*)-RTE-DOC-USER-PARM-1 PIC X(04).
15 (*)-RTE-DOC-USER-PARM-1-FLAG PIC X(01).
88 (*)-C-DOC-USER-PARM-1 VALUE '1'.
88 (*)-C-DOC-NO-USER-PARM-1 VALUE '0'.
15 (*)-RTE-DOC-USER-PARM-2 PIC S9(09) COMP.
15 (*)-RTE-DOC-USER-PARM-2-FLAG PIC X(01).
88 (*)-C-DOC-USER-PARM-2 VALUE '1'.
88 (*)-C-DOC-NO-USER-PARM-2 VALUE '0'.
10 (*)-RTE-DATE-FORMAT PIC X(01).
10 (*)-RTE-TIME-FORMAT PIC X(01).
10 (*)-RTE-RETURN-CODE PIC 9(02).
10 (*)-RTE-MESSAGE-CODE PIC X(08).
```

### Program Processing Logic

The Routing Data exit is called when the document has been stored, registered and prior to being inserted into the AIS+ workflow process. The input parameters are passed as information to the exit and the routing information can be modified. If a field is modified in the exit, change the value of the corresponding flag from 0 to 1. Set the return code field to one of the following values:

**00** - Continue processing.

**01** - Continue processing. Use the values updated by the exit.

**12**, - Stop processing and display a message. The message code that describes the problem is contained in the message code field.

**16**

**20** - A DB2 deadlock or timeout (-911 or -913) was encountered. VSR will perform a restart.

**Note:** When customizing the exit if SQL code is added then you must test for DB2 deadlock and timeout conditions (-911 or -913 sqlcode return). If these conditions are encountered then the exit should send a return code of +20 to indicate the VSR that a retry should be attempted. Failure to test for this condition and return the proper return code will prevent the VSR process from controlling the commit scope of the work.

## Event Format Exit: For FAF Tables

---

The *AIS+ Event Format Exit* is responsible for formatting an event's log entry that can be reviewed using the *AIS+ Document History* function (on-line transaction). This exit is called only when the value of `USEREXIT` is set to "0".

An event log entry represents four (4) lines with each line having a maximum of sixty (60) characters to display (*AIS+ versions 2.0 and earlier*). The first line represents the first 59 characters of the event description and the other three lines represent the event data (the first 180 characters). The exit can build and can send the full 253 characters for both data elements - description and data, but the log will only display as described above. *AIS+ version 2.1* and following allow for larger events. Instead of being limited to four lines of 60, there are now from 1 to 8 lines of 60 available. For a detailed explanation please refer to the *AIS+ User Reference Manual*. From the VSR perspective all events are created the same, regardless of which version of *AIS+* you are currently running. Using the Notes field on the input records, the user can pass information to be used in the comments section of document history. In other words, the values passed in the Notes field can appear as part of a documents event. As noted earlier, the v1.2 and 2.1 input file formats only allow for 120 bytes of Notes, whereas v2.2 allows for 253 bytes. Recall that the first 34 bytes are reserved for system use (see *Special Considerations* above). As such, the maximum bytes allowed for notes is 219. These will be incorporated into event comments. Please note that if you are running *AIS+ v2.0* or earlier, you will not be able to view all of the comments that are generated by these events, even though they exist. *Version 2.1* and beyond of *AIS+* will be able to view all comments that are passed in from the VSR input files.

### Call Statement

The exit will be called for a given event when the event takes place in the registration process. The supplied sample exit (`AIEX014P`) is accessed by the use of a `CALL`. A data area is passed between the exit and the calling program to allow the user to customize the event's description and data areas as needed.

```
CALL - 'AIEX014P' USING
      Event data record (AISW013)
ON EXCEPTION
      exception process
END-CALL.
```

**Default Supplied Formats**

The default event log formats are shown in the following table:

EVENT CODE	FUNCTION	DEFAULT DISPLAY
91	STORE	DOCUMENT SCANNED & STORED FOLDID: xxxxxxxxxxxxxxxxxxxxxxxxxxxx
92	STORE/ROUTE	DOCUMENT SCANNED, STORED, & ROUTED FOLDID: xxxxxxxxxxxxxxxxxxxxxxxxxxxx RTCODE: xxxxxx RTUNIT: xxxx RTLOB: xxxxxx TRANTYPE: xxxxxx HOLD DATE: xxxxxxxxxx HOLD TIME: xxxxxxxxxx
93	DOCUMENT OFF HOLD	OFFHOLD- FOLDID: xxxxxxxxxxxxxxxxxxxxxxxxxxxx OPERID: xxxxxxxx

### Parameter List Structure

The supplied default formats are only a recommendation. You can create your own format for an event's log with the information supplied by AIS+ (through the supplied AISW013 COBOL II copybook). All events will not carry the same information; therefore, you will be limited in what you can build by the information supplied for a given event.

The only requirement the exit must satisfy is that both the event description and the event data must each contain some form of information on returning back to the calling AIS+ function.

The following table contains the information that is available for each event. An **I** in a column represents data that is available to that event for use in formatting a customized log. An **O** in a column represents data that is returned to the calling program.

**NOTE:** If the EVENT DESCRIPTION LENGTH and EVENT DATA LENGTH fields are both set to zero in the return fields, then the event is not created. If events are desired, then at a minimum, the exit must set return values in these fields.

PARAMETER	S T O R E D O C	S T O R E / R O U T E D O C	D O C U M E N T O F F H O L D
TIME FORMAT	I	I	I
DATE FORMAT	I	I	I
EVENT CODE	I	I	I
CREATE SITE	I	I	I
DEST FOLDER ID			I
DEST OBJECT TOKEN			
DEST OBJECT VERS			
DEST RECEIVE DATE			I
FILE DATE	I	I	
FOLDER ID	I	I	I

PARAMETER	S T O R E D O C	S T O R E / R O U T E D O C	D O C U M E N T O F F H O L D
FORM NAME (OLD)	I	I	
MANAGEMENT CLASS			
NUMBER PAGES			
OBJECT DESCR (OLD)	I	I	
OBJECT NAME			
OBJECT TIME			I
OBJECT VERSION			
ORIGINAL KEPT FLAG	I	I	
RECEIVE DATE	I	I	I
RETENTION PERIOD			
SECURITY CLASS (OLD)	I	I	
STORAGE SITE (OLD)	I	I	I
TAB NAME			
TIME CHANGED			
USERID	I	I	I
WORKSTATION ID			I
RLOB		I	I
ROUTING UNIT		I	I



PARAMETER	S T O R E D O C	S T O R E / R O U T E D O C	D O C U M E N T O F F H O L D
TRANSACTION TYPE		I	I
HOLD DATE		I	I
HOLD TIME		I	I
REASON CODE			
REASON MESSAGE			
PRIORITY (OLD)		I	
EXPIRATION DATE (OLD)		I	
REASSIGNMENT ID			I
EVENT DATA	O	O	O
EVENT DATA LENGTH	O	O	O
EVENT DESCRIPTION	O	O	O
EVENT DESCRIPTION LENGTH	O	O	O
COLLECTION NAME	I	I	

## Copy member AISW013

```
03 USEREVNT-PARMS.
05 USEREVNT-COLLNAME PIC X(44).
05 USEREVNT-COMMENT1 PIC X(60).
05 USEREVNT-COMMENT2 PIC X(60).
05 USEREVNT-COPYTYPE PIC X(01).
05 USEREVNT-CRTEsite-FROM PIC X(04).
05 USEREVNT-CRTEsite-TO PIC X(04).
05 USEREVNT-DATE-FORMAT PIC X(01).
05 USEREVNT-DFOLDID PIC X(40).
05 USEREVNT-DOBJTKN.
10 USEREVNT-DOBJTIME PIC X(26).
10 USEREVNT-DCRTEsite PIC X(04).
05 USEREVNT-DOBJVERS PIC S9(4) COMP.
05 USEREVNT-DRECVDT PIC X(10).
05 USEREVNT-DTIMECHG PIC X(26).
05 USEREVNT-EVNTCD PIC S9(4) COMP.
05 USEREVNT-D-EVNTDATA PIC X(253).
05 USEREVNT-L-EVNTDATA PIC S9(9) COMP.
05 USEREVNT-D-EVNTDESC PIC X(253).
05 USEREVNT-L-EVNTDESC PIC S9(9) COMP.
05 USEREVNT-EXPIRE-DATE.
10 USEREVNT-EXPIRE-DATE-FROM PIC X(10).
10 USEREVNT-EXPIRE-DATE-TO PIC X(10).
05 USEREVNT-FILEDATE PIC X(10).
05 USEREVNT-FOLDID PIC X(40).
05 USEREVNT-FORMNUM-FROM PIC X(16).
05 USEREVNT-FORMNUM-TO PIC X(16).
05 USEREVNT-HOLD-DATE PIC X(10).
05 USEREVNT-HOLD-TIME PIC X(08).
05 USEREVNT-MGMTCL PIC X(08).
05 USEREVNT-NRECVDT PIC X(10).
05 USEREVNT-NUMPAGES PIC S9(4) COMP.
05 USEREVNT-OBJDESC-FROM PIC X(253).
05 USEREVNT-OBJDESC-TO PIC X(253).
05 USEREVNT-OBJNAME PIC X(253).
05 USEREVNT-OBJTIME PIC X(26).
05 USEREVNT-OBJVERS PIC S9(4) COMP.
05 USEREVNT-ORIGKEPT PIC X(01).
05 USEREVNT-PAGE-AFTER PIC S9(04) COMP.
05 USEREVNT-PAGE-NUM PIC S9(04) COMP.
05 USEREVNT-PAGE-START PIC S9(04) COMP.
05 USEREVNT-PRIORITY.
10 USEREVNT-PRIORITY-FROM PIC S9(4) COMP.
10 USEREVNT-PRIORITY-TO PIC S9(4) COMP.
05 USEREVNT-PRTALT PIC X(01).
05 USEREVNT-PRTID PIC X(08).
05 USEREVNT-RCODE PIC X(06).
05 USEREVNT-REASON-CD PIC X(02).
05 USEREVNT-REASON-MSG PIC X(50).
05 USEREVNT-REASSIG-ID PIC X(08).
05 USEREVNT-RECVDATE PIC X(10).
05 USEREVNT-RETPER PIC S9(9) COMP.
05 USEREVNT-RLOB PIC X(06).
05 USEREVNT-RUNIT PIC S9(9) COMP.
05 USEREVNT-SECURCL-FROM PIC X(02).
05 USEREVNT-SECURCL-TO PIC X(02).
05 USEREVNT-STGCL PIC X(08).
05 USEREVNT-STORSITE-FROM PIC X(04).
05 USEREVNT-STORSITE-TO PIC X(04).
05 USEREVNT-TAB PIC X(16).
05 USEREVNT-TERMID PIC X(08).
05 USEREVNT-TIMECHGD PIC X(26).
```

```
05 USEREVNT-TIME-FORMAT PIC X(01).
05 USEREVNT-TRANTYPE PIC X(06).
05 USEREVNT-USERID PIC X(08).
05 USEREVNT-USERSTAT PIC X(01).
05 USEREVNT-VCOLLNAM PIC X(44).
05 USEREVNT-VEREXIST PIC S9(4) COMP.
05 USEREVNT-VMGMTCL PIC X(08).
05 USEREVNT-VRETPER PIC S9(9) COMP.
05 USEREVNT-VSTGCL PIC X(08).
05 USEREVNT-VSTORSIT PIC X(04).
05 USEREVNT-WKSTATID PIC X(08).
05 USEREVNT-FROM-TAB-NAME PIC X(16).
05 USEREVNT-TO-TAB-NAME PIC X(16).
```

## Customer Exit Locations and Function Codes: For FAF Tables

At certain points during processing, the AIS+ system passes control to the user exits provided by your enterprise. The exit identifies the call by means of the function codes and option codes. The function code is a 2-character code specifying the function that calls the exit. The option codes range from 1 to 5 for the Customer Data exit.

The following table shows which VSR functions call the user exits and the purpose of calling the user exit from that function. It also identifies the option codes for each function.

<b>Customer Exit Locations and Option Codes</b>			
<b>User Exit</b>	<b>Function</b>	<b>Option Code</b>	<b>Purpose</b>
Customer Data Exit (AIEX007P)	Folder Addition in Update/Registration	5	Changes folder information.
	Retrieve UserParm1 and UserParm2 in Update/Registration	1	When routing a document this user exit will be called to obtain the user parameter 1 and user parameter 2 fields to determine the unit code and route code.
Data Collection Exit (AIEX008P)	Document Validation	None	Changes the document information of a document before placing the document in the VSR process.
Routing Exit (AIEX006P)	Route Document in Update/Registration	None	Changes any of the routing information before placing the document in the routing queue. AIS Plus calls the routing data exit after the document is stored.
Event Format Exit (AIEX014P)	Document off hold in Update/Registration	None	Allows for the changing of the event data format before the creation of an event. If the event description and data description length fields are returned with zero then the event is not created.
	Document scanned and stored	None	Allows for the changing of the event data format before the creation of an event. If the event description and data description length fields are returned with zero then the event is not created.
	Document scanned, stored, and routed	None	Allows for the changing of the event data format before the creation of an event. If the event description and data description length fields are returned with zero then the event is not created.

## Functions and Event Codes: For FAF Tables

---

This table lists the function codes defined for the AIS+ online functions. The function codes are passed to the user exits to identify what function the exit is being called from. Event history records are created for some functions. This is indicated in the following table. The event code identifies the event history message text associated with the function performed.

### Off-line Exit Function Codes

This table provides the function codes for the online AIS+ system that indicate if an event is recorded for a function.

Off-line Exit Function Codes		
Function	Function Code	Event Record
VSR document	91	Yes
VSR and route document	92	Yes
VSR document(s) off hold	93	Yes

## Custom Data Validation Exit: For Non-FAF Implementation

This section explains how to customize the Custom Data Validation exit. The use of the exit is controlled by the USEREXIT flag in the control file (set flag = "1"). The call to the exit takes place before the document has been stored. The Custom Validation exit:

Provides an opportunity for the user to implement a custom validation process for object storage.

Allows for the rejection of objects to the image reject file.

### Call Statement

The call statement used for the Custom Data Validation exit is as follows:

```
CALL'AIEX009P' USING
      VSR Control file (AIBSCNTL)
      VSR VALIN 2.2 format index record (AIBSV22I)
      VSR Custom Exit Control Structure (AIBSCEXC)
ON EXCEPTION
      exception process
END-CALL.
```

### Parameter List Structure

The call to the Custom Data Validation Exit uses the control file, the ODM version 2.2 VALIN index record, and a customer exit control structure as input. These values are made available for custom validation process, including changing values and passing these values back to the OAM store and custom registration processes. Use the customer exit control structure to populate and manipulate the object name. If the object name is not changed, the standard FAF naming convention (based on timestamp) will be used.

The input parameters are passed as information to the exit. If the exit modifies a field, it changes the value of the corresponding field, and sets the control structure return code to 1 to indicate changed values. The exit can also set values to indicate that the record should be rejected, and that the process should be halted. These values are as follows:

**00** - Continue processing.

**01** - Continue processing. Use the values updated by the exit.

**12** - Reject the record input for validation to the reject file.

**16** - Stop the process. The control structure allows the custom exit to return a message for display on the error reports.

**20** - DB2 has encountered a deadlock or timeout (-911 or -913). VSR will perform a restart.

**Note:** When customizing the exit if SQL code is added then you must test for DB2 deadlock and timeout conditions (-911 or -913 sqlcode return). If these conditions are encountered then the exit should send a return code of +20 to indicate the VSR that a retry should be attempted. Failure to test for this condition and return the proper return code will prevent the VSR process from controlling the commit scope of the work.

## Custom Data Registration Exit: For Non-FAF Implementation

This section explains how to customize the Custom Data Registration exit. The use of the exit is controlled by the USEREXIT flag in the control file (set flag = "1"). The call to the exit takes place after the document has been stored. The Custom Registration exit:

- provides an opportunity for the user to implement a custom registration process for object storage; and
- allows for the rejection of objects to the image reject file.

### Call Statement

The call statement used for the Custom Data Registration exit is as follows:

```
CALL'AIEX010P' USING
      VSR Control file (AIBSCNTL)
      VSR VALIN 2.2 format OAM index record (AIBSOAMI)
      VSR Custom Exit Control Structure (AIBSCEXC)
ON EXCEPTION
      exception process
END-CALL.
```

### Parameter List Structure

The call to the Custom Data Registration Exit uses both the control file and the ODM version 2.2 OAM index record as input. These values are made available for custom registration processing.

The input parameters are passed as information to the exit. If the exit modifies a field, it changes the value of the corresponding field, and sets the control structure return code to 1 to indicate changed values. The exit can also set values to indicate that the record should be rejected, and that the process should be halted. These values are as follows:

- 00** - Continue processing.
- 01** - Continue processing. Use the values updated by the exit.
- 16** - Stop the process. The control structure allows the custom exit to return a message for display on the error reports.
- 20** - DB2 has encountered a deadlock or timeout (-911 or -913). VSR will perform a restart.

**Important Note:** Because the object has been stored into OAM before registration takes place, if image registration cannot process successfully, the process should be halted. If the process is not halted, the custom registration results may be unpredictable and the images may not be retrieved after storing the object.

When customizing the exit if SQL code is added then you must test for DB2 deadlock and timeout conditions (-911 or -913 sqlcode return). If these conditions are encountered then the exit should send a return code of +20 to indicate the VSR that a retry should be attempted. Failure to test for this condition and return the proper return code will prevent the VSR process from controlling the commit scope of the work.

## VSR Error Messages and Recovery

---

Errors VSR encounter when processing is logged to an error report (ERRRLOG1) generated during program execution. In addition, ERRRLOG2 may contain some additional error information from the exit programs. WORKRPT1 and WORKRPT2 contain run statistics and error eye catcher messages.

The following return codes are possible from the program execution:

- 00 - Successful run, no errors or warnings encountered.
- 01 - Run completed, the input data file was empty.
- 04 - Run completed with warnings. See error reports ERRRLOG1 and ERRRLOG2 for possible messages.
- 08 - Run completed with rejected record(s). See error reports ERRRLOG1 and ERRRLOG2 for possible messages. See also the ERRRDAT file for output of the error records in the input format. If a record is rejected and a DB2 deadlock or timeout is encountered the VSR -911 restart process internal to the program cannot be executed. The rejected record(s) can be corrected and reentered.
- 12 - An error was encountered in the validation of the parameters in the control file (CNTRLDAT) or in the reading of the AISRCOV table. See error reports ERRRLOG1 and WORKRPT1 for possible messages. These errors must be corrected before the process will execute.
- 16 - A fatal error was encountered in processing. See error reports ERRRLOG1, ERRRLOG2, and work reports WORKRPT1 and WORKRPT2 for possible messages. This is a severe failure and the process is halted after performing a rollback to the last commit point in the processing.

For VSR to run successfully, some default parameters must be pulled from a row of the AIS recovery table, AISRCOV. Therefore, a row must be inserted into table AISRCOV that contains the vital run information described in the following section. The default run information is pulled from a row with the following key values:

```

FIELD NAME VALUE
RCOV_PROG_NAME AIBS001P
RCOV_STEP_NAME AIBS001P
RCOV_JOB_NAME - specified in the CNTL file/ defaulted to AIBS001P
RCOV_CNTRL_INTERNAL DEFAULTRUNPARMS

```

It should be noted that, because RCOV\_JOB\_NAME may be specified within the CNTL file, it is possible for a customer to have multiple default rows within the AISRCOV table. However, at a bare minimum, one default row with 'AIBS001P' within RCOV\_JOB\_NAME column needs to be inserted into the AISRCOV table.

The generation of the AISRCOV record is a site specific task. DB2 SPUFI inserts or other DB2 data manipulation tools can be used for this. The AISRCOV key field settings for the default row for VSR processing are:



FIELD NAME VALUE

RCOV\_PROG\_NAME AIBS001P

RCOV\_STEP\_NAME AIBS001P

RCOV\_JOB\_NAME AIBS001P

RCOV\_CNTL\_INTERNAL DEFAULTRUNPARMS

The following AISRCOV fields are used by AIBS001P and should be set with values:

FIELD NAME VALUE

RCOV\_EFF\_PEAK\_TIME Any time after this time and before RCOV\_EFF\_BTCH\_TIME is considered peak processing time, and commits are performed based on RCOV\_PEAK\_CMIT\_ROWS value.

RCOV\_PEAK\_CMT\_ROWS Number of rows updated between commit points.

RCOV\_EFF\_BTCH\_TIME Any time after this time and before RCOV\_EFF\_PEAK\_TIME is considered non-peak processing time, and commits are performed based on RCOV\_BAT\_CMIT\_ROWS value.

RCOV\_BAT\_CMT\_ROWS Number of rows updated between commit points in non-peak processing time.

RCOV\_MAX\_RTRY\_911 Number of retries for deadlock before generating errors.

RCOV\_LOW\_KEY Should be set to high values.

RCOV\_HIGH\_KEY Should be set to high values.

Additional default run parameter rows can be added to the AISRCOV table by duplicating the above insert with a unique RCOV\_JOB\_NAME. This row may then be utilized by specifying the unique Job Name within the VSR control file.

At the user specified commit points within the default run parameter row, a new AISRCOV row will be inserted with the appropriate key information to allow for program restart. This row will contain unique keys to allow for start of another batch run while a restarted run continues. In the event of a restart/recovery, the program will read AISRCOV, and will begin processing at the next record after the record key information found in the recovery table. If the recovery table entry is not found the program will issue an error message which will be print on the error report and immediately end processing. In the event of a system failure, partial error and results reports may be generated. These may be kept depending on JCL DISP parameters, and should be set according to site specific guidelines.

RCOV\_JOB\_TERM\_FL Set to "C" for successful completion of run.

The AISRCOV key field settings for the restart row for VSR processing are:

FIELD NAME VALUE

RCOV\_PROG\_NAME AIBS001P

RCOV\_STEP\_NAME CURRENT TIME

RCOV\_JOB\_NAME CURRENT DATE

RCOV\_CNTL\_INTERNAL Batch run ID from the control file

See Appendix A for an explanation of the AISRCOV Table.

In the event of a recovery processing will begin at the record indicated within the RCOV\_HIGH\_KEY\_TEXT column.

Upon successful completion of a VSR run, the recovery table entry created will be deleted or kept based on the KEEPRCOV parameter in the CNTRLDAT file. Once a run is marked complete the record cannot be used to restart that particular batch run. If batch Ids are not unique then the user must enter a combination of RCOV\_STEP\_NAME, RCOV\_JOB\_NAME, and RCOV\_CNTL\_INTERNAL in the JCL stream for a restart process. This ensures that subsequent runs of the process will not reprocess data erroneously. If unique Batch Run Id's are used for every run, then only that Batch Run Id needs to be specified for a restart.

Please note that if an error occurs before any records were processed that it is not considered a restart scenario and no row is inserted into the AISRCOV table. After the error has been rectified, the job that failed should be run as if no error was encountered at all. Typically, this error will be due to such things as invalid CNTL parameter values or inaccessible resources (DB2).

## Installation of VSR Software

This section provides sample instructions for the execution of AIS+ VSR. Additionally, instructions are presented for running the process in a number of scenarios.

Use the following sample JCL to unload tape files. This JCL is provided in data set BSSV30.UNLOAD.JCL. You may need to customize the JCL to match your site's standards.

```
//???????? JOB (),'IEBCOPY AISV30 3480',
//          CLASS=?,MSGCLASS=?,MSGLEVEL=(1,1),
//          NOTIFY=??????
//*
//*****
//* 1. MODIFY THE JOBCARD
//* 2. THERE ARE 6 PARTITIONED DATASETS CREATED (DISKLIB=) :
//*     VSRV30.SRCLIB
//*     VSRV30.DBRMLIB
//*     VSRV30.DTALIB
//*     VSRV30.CPYLIB
//*     VSRV30.OBJLIB
//*     VSRV30.JCLLIB
//* THE ABOVE DATASET NAMES THAT ARE ASSIGNED TO DISKLIB CAN
//* BE CHANGED ACCORDING TO YOUR SITE'S STANDARDS.
//* 3. MODIFY 'UNIT=(TAPE,' TO MATCH YOUR SITE STANDARDS
//* 4. SPACE IS ALLOCATED IN BLOCKS USING BLKSIZE. REQUIREMENTS
//* IN CYL/TRK ARE LISTED BY EACH LIBRARIES SPACE SYMBOLICS
//* 5. JOB IS RESTARTABLE IN ANY STEP
//*
//*****
//* UNLOAD THE PDS MEMBERS THAT CONTAIN VSR MODULES
//*****
//*
//INSTLIBS PROC TAPE=TAPE,
//          WRKUNIT=SYSDA,
//          LIBUNIT=SYSDA,
//          VOLSER=VSR30,
//          OUT='*',
//          PRIQTY=1,SECQTY=1,DIRQTY=1, /* SPACE IS IN BLOCKS */
//          RECFM=FB,LRECL=80,BLKSIZE=3120,
//          DISKLIB=,TAPELIB=
//*
//ICOPY EXEC PGM=IEBCOPY
//TAPELIB DD DSN=&TAPELIB,DISP=OLD,LABEL=(&FILENUM,SL),
//          UNIT=(&TAPE,,DEFER),
//          VOL=(,RETAIN,SER=&VOLSER)
//DISKLIB DD DSN=&DISKLIB,DISP=(,CATLG,DELETE),
//          SPACE=(&BLKSIZE,(&PRIQTY,&SECQTY,&DIRQTY),,ROUND),
//          UNIT=&LIBUNIT,
//          DCB=(DSORG=PO,RECFM=&RECFM,LRECL=&LRECL,BLKSIZE=&BLKSIZE)
//*
//SYSUT3 DD SPACE=(CYL,(1,1)),UNIT=&WRKUNIT
//SYSUT4 DD SPACE=(CYL,(1,1)),UNIT=&WRKUNIT
//SYSPRINT DD SYSOUT=&OUT
//SYSIN DD DUMMY
// PEND /* END OF INSTLIBS */
//*
//SRCLIB EXEC INSTLIBS,FILENUM=2,
//          PRIQTY=250,SECQTY=250,DIRQTY=30,
//          DISKLIB='VSRV30.SRCLIB', /*DDNAME CAN BE CHANGED*/
//          TAPELIB='BSSV30.SRCLIB'
//SYSIN DD *
COPY INDD=TAPELIB,OUTDD=DISKLIB
//*
//DBRM EXEC INSTLIBS,FILENUM=3,
//          PRIQTY=500,SECQTY=250,DIRQTY=60,
//          DISKLIB='VSRV30.DBRMLIB', /*DDNAME CAN BE CHANGED*/
//          TAPELIB='BSSV30.DBRMLIB'
//SYSIN DD *
```

```

COPY INDD=TAPELIB,OUTDD=DISKLIB
//*
//DATA EXEC INSTLIBS,FILENUM=4,
// PRIQTY=500,SECQTY=250,DIRQTY=30,
// DISKLIB='VSRV30.DTALIB', /*DDNAME CAN BE CHANGED*/
// TAPELIB='BSSV30.DTALIB'
//SYSIN DD *
COPY INDD=TAPELIB,OUTDD=DISKLIB
//*
//CPY EXEC INSTLIBS,FILENUM=5,
// PRIQTY=250,SECQTY=250,DIRQTY=7,
// DISKLIB='VSRV30.CPYLIB', /*DDNAME CAN BE CHANGED*/
// TAPELIB='BSSV30.CPYLIB'
//SYSIN DD *
COPY INDD=TAPELIB,OUTDD=DISKLIB
//*
//OBJ EXEC INSTLIBS,FILENUM=6,
// PRIQTY=250,SECQTY=250,DIRQTY=7,
// DISKLIB='VSRV30.OBJLIB', /*DDNAME CAN BE CHANGED*/
// TAPELIB='BSSV30.OBJLIB'
//SYSIN DD *
COPY INDD=TAPELIB,OUTDD=DISKLIB
//*
//JCL EXEC INSTLIBS,FILENUM=7,
// PRIQTY=250,SECQTY=250,DIRQTY=7,
// DISKLIB='VSRV30.JCLLIB', /*DDNAME CAN BE CHANGED*/
// TAPELIB='BSSV30.JCLLIB'
//SYSIN DD *
COPY INDD=TAPELIB,OUTDD=DISKLIB
//
//

```

Modify and use the provided sample JCL, VSRV30.JCLLIB(VSRASML), to linkedit all programs. AIBS700P, AIBS701P, and AIBS702P are assembler programs. AIBS001P, AIBS002P, AIBS003P, AIFP001P, AIDTDRVR, AIEX006P, AIEX007P, AIEX008P, AIEX009P, AIEX010P, and AIEX014P are COBOL programs. (Programs beginning with 'AIEX' are stub exit programs that can be customized to meet your site requirements.)

```

//????????? JOB (????,????), '@@@@@@',
// CLASS=?,MSGCLASS=?,NOTIFY=??????
//
//JOB LIB DD DSN=SYS1.V2R3.DSNLOAD,DISP=SHR
//
//*****
//*
//* SAMPLE JOB TO LINKEDIT ALL OF YOUR BATCH PROGRAMS
//*
//* ADD YOUR JOBCARD
//*
//* MODIFY THE OVERRIDES ON THE PROC STATEMENT TO MATCH YOUR
//* DEFINED LIBRARIES
//*
//*****
//*
//LNKBTCH PROC MODULE='PROGRAM',
// REG=1024K,
// LKEDPARM='XREF,LET,LIST,MAP,AMODE(31),RMODE(ANY)',
// COB2LIB='SYS1.COB2LIB',
// DSNLOAD='SYS1.V2R3.SDSNLOAD',
// DSNEXIT='SYS1.V2R3.SDSNEXIT',
// CSSLIB='SYS1.CSSLIB',
// EYPLMD1='SYS1.SEYPLMD1',
// LOADLIB='AIS.VSRV30.LOADLIB',
// AISHLQ='AIS.VSRV30',
// OUTC='*'
//*
//LKED EXEC PGM=IEWL,REGION=&REG,PARM='&LKEDPARM'
//SYSLIB DD DSN=&COB2LIB,DISP=SHR

```

```

//      DD DSN=&DSNLOAD,DISP=SHR
//      DD DSN=&DSNEXIT,DISP=SHR
//      DD DSN=&CSSLIB,DISP=SHR
//      DD DSN=&EYPLMD1,DISP=SHR
//      DD DSN=&LOADLIB,DISP=SHR
//SYSLIN DD DSN=&AISHLQ.OBJLIB(&MODULE),DISP=SHR
//      DD DDNAME=SYSIN
//SYSLMOD DD DSN=&LOADLIB(&MODULE),DISP=SHR
//SYSUT1 DD SPACE=(1024,(50,50)),UNIT=SYSDA
//SYSPRINT DD SYSOUT=&OUTC
// PEND
//*****
// * LINKEDIT FOR AIBS700P
//*****
//LINK700P EXEC LNKBTCH,
//      MODULE=AIBS700P
// *
//*****
// * LINKEDIT FOR AIBS701P
//*****
//LINK701P EXEC LNKBTCH,
//      MODULE=AIBS701P
// *
//*****
// * LINKEDIT FOR AIBS702P
//*****
//LINK702P EXEC LNKBTCH,
//      MODULE=AIBS702P
// *
//*****
// * LINKEDIT FOR AIDTDRVR
//*****
//LINKDRVR EXEC LNKBTCH,
//      MODULE=AIDTDRVR
// *
//*****
// * LINKEDIT FOR AIEX006P
//*****
//LINKE6P EXEC LNKBTCH,
//      MODULE=AIEX006P
// *
//*****
// * LINKEDIT FOR AIEX007P
//*****
//LINKE7P EXEC LNKBTCH,
//      MODULE=AIEX007P
// *
//*****
// * LINKEDIT FOR AIEX008P
//*****
//LINKE8P EXEC LNKBTCH,
//      MODULE=AIEX008P
// *
//*****
// * LINKEDIT FOR AIEX009P
//*****
//LINKE9P EXEC LNKBTCH,
//      MODULE=AIEX009P
// *
//*****
// * LINKEDIT FOR AIEX010P
//*****
//LINK10P EXEC LNKBTCH,
//      MODULE=AIEX010P
// *
//*****
// * LINKEDIT FOR AIEX014P
//*****
//LINK14P EXEC LNKBTCH,
//      MODULE=AIEX014P
// *
//*****

```

```
/* LINKEDIT FOR AIFP001P
/*****
//LINKFP1P EXEC LNKBTCH,
//      MODULE=AIFP001P
//LKED.SYSIN DD *
INCLUDE SYSLIB(DSNALI)
/*
/*****
/* LINKEDIT FOR AIBS001P
/*****
//LINKBS1P EXEC LNKBTCH,
//      MODULE=AIBS001P
//LKED.SYSIN DD *
INCLUDE SYSLIB(DSNALI)
/*
/*****
/* LINKEDIT FOR AIBS002P
/*****
//LINKBS2P EXEC LNKBTCH,
//      MODULE=AIBS002P
/*
/*****
/* LINKEDIT FOR AIBS003P
/*****
//LINKBS3P EXEC LNKBTCH,
//      MODULE=AIBS003P
/*
//
```

Modify JCL VSR30.JCLLIB(BINDVSR) to conform to your sites standards and to include the DBRM modules required to support your storage groups (CBRISS00 for storage groups 00 through 04, CBRISS05 for storage groups 05 through 09, etc.). Submit this job to bind the VSR DB2 plan.

```
//????????? JOB (????,????), '@@@@@@', CLASS=?, MSGCLASS=?, NOTIFY=????????,
//      REGION=1024K
// *
//*****
//***** CHANGE ALL ? AND @ ABOVE *
//***** CHANGE ALL &&OAMDBRM TO YOUR SYSTEM DBRM LIBRARY THAT *
//***** CONTAINS THE CBRNNNNN MEMBERS *
//***** CHANGE ALL &&HIQUAL TO YOUR VSR HIGH LEVEL QUALIFIER *
//***** CHANGE ALL &&HIQUAL2 TO YOUR AIS+ HIGH LEVEL QUALIFIER *
//*****
// *
//JOBLIB DD DSN=AISDB2.DSNLOAD,DISP=SHR
//BINDSTEP EXEC PGM=IKJEFT01,DYNAMNBR=20
//DBRMLIB DD DSN=&&OAMDBRM.DBRMLIB,DISP=SHR
//      DD DSN=&&HIQUAL.DBRMLIB,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD *
      DSN SYSTEM(DB2S)
      BIND PLAN(AIBS001P) -
          MEM(AIBS001P,AIBS002P,AIBS003P,AIEX006P,AIEX007P,AIEX008P, -
              AIEX009P,AIEX010P,AIDTDRVR,CBRHTBSV,CBRISS00,CBRISS05) -
          ACT (REP) -
          OWNER (AISADM) -
          RETAIN -
          ISOLATION (CS) -
          VALIDATE (BIND) -
          ACQUIRE (ALLOCATE) -
          RELEASE (DEALLOCATE) -
          EXPLAIN (NO)
      BIND PLAN(AIDTDRVR) -
          MEM(AIDTDRVR) -
          ACT (REP) -
          OWNER (AISADM) -
          RETAIN -
          ISOLATION (CS) -
          VALIDATE (BIND) -
          ACQUIRE (ALLOCATE) -
          RELEASE (DEALLOCATE) -
          EXPLAIN (NO)
      END
// *
// *
//
```

Insert a row into the AIS+ recovery table for default VSR run parameters. These parameters will stay stagnant for every run of the VSR process. The sample SPUFI member INSRCOV in the VSR JCL library should be used to insert the row. If you are not running AIS+, please refer to Appendix A of this manual for information concerning the creation of the recovery table.

## Storing Objects using VSR

Use the sample control file located in the VSR source library ('VSR30.DTALIB(CNTLPARM)') and update the parameters for your installation. For normal runs of VSR, the defaults provided for the run control should be used. Once the control file has been updated, update JCL member AIBS01P in the VSR30.JCLLIB library, changing variables to match your installation specific values. The following is the jobstream you will be modifying:

```
//????????? JOB (????,????), '@@@@@@', CLASS=?,MSGCLASS=?, 0001000
// NOTIFY=????????? 0001000
//*
//*****
//***** CHANGE ALL ? AND @ ABOVE *
//***** CHANGE ALL &&DB2SYS BELOW TO YOUR INSTALLATIONS SYBSYSTEM ID*
//***** CHANGE ALL &&DB2.DSNLOAD TO YOUR DB2 SYSTEM LOADLIB *
//***** CHANGE ALL &&SITEHLQ TO YOUR VSR HIGH LEVEL QUALIFIER *
//***** CHANGE ALL &&MSGCLASS TO YOUR DESIRED OUTPUT CLASS OR '*' *
//***** CHANGE ALL &&UNIT TO YOUR INSTALLATIONS DASD UNIT *
//***** CHANGE ALL &&HSCS TO YOUR HSCS FILE NAME EXTENSIONS *
//*****
//DATEDRV PROC HIQUAL='&&SITEHLQ',
// MSGOUT='&&MSGCLASS'
// UNIT='&&UNIT'
// HSCS='&&HSCS'
//*
//=====
//STEP001 EXEC PGM=AIDTDRVR,DYNAMNBR=30,
// TIME=1440,REGION=32M
//=====
//STEPLIB DD DSN=&&DB2.DSNLOAD,DISP=SHR
//SYSOUT DD SYSOUT=&MSGOUT
//SYSABOUT DD SYSOUT=&MSGOUT
//SYSUDUMP DD SYSOUT=&MSGOUT
//SYSTSPRT DD SYSOUT=&MSGOUT
//SYSPRINT DD SYSOUT=&MSGOUT
//DTFILE DD DSN=&&DTPASED,DISP=(MOD,PASS),
// UNIT=SYSDA,SPACE=(80,(5,10),RLSE),
// DCB=(RECFM=FB,LRECL=80)
//DATEDRV PEND /**** END OF PROC DATEDRV ****/
//*
//DATERUN EXEC DATEDRV
//*
//VSRPROC PROC HIQUAL='&&SITEHLQ',
// MSGOUT='&&MSGCLASS'
// UNIT='&&UNIT'
// HSCS='&&HSCS'
//*
//STEP001 EXEC PGM=IEFB14
//SYSPRINT DD SYSOUT=&MSGOUT
//ERRRLOG1 DD DSN=&HIQUAL.VSRPROD.ERLOG1,
// DISP=(MOD,DELETE,DELETE),UNIT=&UNIT,SPACE=(TRK,0)
//ERRRLOG2 DD DSN=&HIQUAL.VSRPROD.ERLOG2,
// DISP=(MOD,DELETE,DELETE),UNIT=&UNIT,SPACE=(TRK,0)
//WORKRPT1 DD DSN=&HIQUAL.VSRPROD.WRKR1,
// DISP=(MOD,DELETE,DELETE),UNIT=&UNIT,SPACE=(TRK,0)
//WORKRPT2 DD DSN=&HIQUAL.VSRPROD.WRKR2,
// DISP=(MOD,DELETE,DELETE),UNIT=&UNIT,SPACE=(TRK,0)
//OUTDATA DD DSN=&HIQUAL.VSRPROD.OUTDAT,
// DISP=(MOD,DELETE,DELETE),UNIT=&UNIT,SPACE=(TRK,0)
//ERRDATA DD DSN=&HIQUAL.VSRPROD.ERRDAT,
// DISP=(MOD,DELETE,DELETE),UNIT=&UNIT,SPACE=(TRK,0)
//*
//STEP002 EXEC PGM=AIBS001P,COND=(0,NE),DYNAMNBR=30,
// TIME=1440,REGION=30M
```



```
//STEPLIB DD DSN=&&DB2.DSNLOAD,DISP=SHR
//SYSOUT DD SYSOUT=&MSGOUT
//SYSABOUT DD SYSOUT=&MSGOUT
//SYSUDUMP DD SYSOUT=&MSGOUT
//SYSTSPRT DD SYSOUT=&MSGOUT
//SYSPRINT DD SYSOUT=&MSGOUT
//ERRRLOG1 DD DSN=&HIQUAL..VSRPROD.ERLOG1,DISP=(,CATLG,KEEP),
//  UNIT=&UNIT,SPACE=(133,(25,10),RLSE),
//  DCB=(RECFM=FB,LRECL=133)
//ERRRLOG2 DD DSN=&HIQUAL..VSRPROD.ERLOG2,DISP=(,CATLG,KEEP),
//  UNIT=&UNIT,SPACE=(133,(25,10),RLSE),
//  DCB=(RECFM=FB,LRECL=133)
//WORKRPT1 DD DSN=&HIQUAL..VSRPROD.WRKRPT1,DISP=(,CATLG,KEEP),
//  UNIT=&UNIT,SPACE=(133,(25,10),RLSE),
//  DCB=(RECFM=FB,LRECL=133)
//WORKRPT2 DD DSN=&HIQUAL..VSRPROD.WRKRPT2,DISP=(,CATLG,KEEP),
//  UNIT=&UNIT,SPACE=(133,(25,10),RLSE),
//  DCB=(RECFM=FB,LRECL=133)
//OUTDATA DD DSN=&HIQUAL..VSRPROD.OUTDAT,DISP=(NEW,CATLG,KEEP),
```

```

// UNIT=&UNIT,AVGREC=K,SPACE=(CYL,(5,5),RLSE),
// DCB=(RECFM=VB,LRECL=32750,BLKSIZE=32754)
//ERRDATA DD DSN=&HIQUAL..VSRPROD.ERRDAT,DISP=(NEW,CATLG,KEEP),
// UNIT=&UNIT,AVGREC=K,SPACE=(CYL,(25,20),RLSE),
// DCB=(RECFM=VB,LRECL=32750,BLKSIZE=32754)
//CNTRLDAT DD DSN=&HIQUAL..SYSCOMVSR.PARMLIB(CNTLPARM),DISP=SHR
//INDATA1 DD DSN=&HIQUAL..&HSCS,DISP=SHR,
// DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760) 00100095
//VSRPROC PEND /***** END OF PROC VSRPROC *****/
//*
//*****
//VSRRUN EXEC VSRPROC
//
//*
//*****
/** IF YOU ARE RESTARTING DUE TO ERRORS, PLACE THE PROPER INFORMATION *
/** IN THE PARMS BELOW AND REPLACE STEP002 EXEC STATEMENT ABOVE. *
/** BBBBBBBBBBBBBBBBBBBB = THE BATCH RUN ID OF THE ORIGINAL RUN FROM *
/** THE CONTROL FILE *
/** DDDDDDDD = A RESTART DATE - THIS VALUE IS FOUND IN THE *
/** RECOVERY TABLE IN COLUMN RCOV_JOB_NAME *
/** FORMAT IS YYYYMMDD *
/** TTTTTTTT = A RESTART TIME - THIS VALUE IS FOUND IN THE *
/** RECOVERY TABLE IN COLUMN RCOV_STEP_NAME *
/** FORMAT IS HH:MM:SS *
//*****
//STEP002 EXEC PGM=AIBS001P,COND=(0,NE),DYNAMNBR=30,
// TIME=1440,REGION=30M,
// PARM='BBBBBBBBBBBBBBBBBBB DDDDDDDD TTTTTTTT'

```

**Note:**

If your site creates unique batch IDs for each run then only this batch identifier is needed in the event of a restart. In this case the parm card would be:

```
// PARM='BBBBBBBBBBBBBBBBBBB'
```

where "BBB..." is the unique batch ID used in the incomplete run. If the batch IDs are not unique and the additional date and time parameters are not entered the restart will not process.

## Appendix A: AISRCOV Table

Certain AIS+ on-line and batch processes require a facility to either initiate a function with specific information, or to allow recovery when a failure has been detected.

There is no current on-line procedure to insert entries into the table. A DB2 utility or tool will be required to populate the table with information required to execute the specific function.

The following table contains the naming conventions used for the AIS+ Recovery/Restart table.

DB2 Component	AIS+ Supplied Name	Your Company Naming Convention
STORAGE GROUP	N/A	
DATABASE	N/A	
TABLESPACE	AISTSRV	
TABLE	AISRCOV	
INDEX	AISIXRV1	

Insertions and modifications to the table are extremely frequent. A single unique index is supplied (but not required) that uses the program name, the step name, and the job name, all in ascending order.

Each AIS+ AISRCOV data entry has a length of 4047 bytes. Each index entry has a length of 24 bytes.

AISRCOV utilizes a simple single tablespace. This table is constantly updated, so set the locking mechanism to use "page" (though most shops will utilize "any"). Since the size of the record consumes a page, percent free is set to zero (0) and free pages set to zero (0). There is no special requirement to have the table reside in any special bufferpool. The data does not need to be erased when the table is dropped. The application frequently uses this table, so do not close the tablespace each time it is used.

Utilizing the space calculations as prescribed in the IBM DB2 Administration Guide V2R3 pages 2-37 through 2-40, the AISRCOV table space calculation is as follows:

$$\begin{aligned}
 \text{usable page size} &= (\text{page size} - 22) * ((100 - \text{pctfree})/100) \\
 \text{records per page} &= \text{FLOOR}(\text{usable page size}/(\text{record size} + \text{overhead})) \\
 \text{pages used} &= 2 + \text{CEILING}(\text{number of rows}/\text{records per page}) \\
 \text{total pages} &= \text{FLOOR}(\text{pages used} * ((1 + \text{freepage})/\text{freepage})) \\
 \text{est. kilobytes} &= \text{total pages} * 4 \\
 \text{where: pages size} &= 4k \text{ pctfree} = 0 \\
 \text{record size} &= 4047 \text{ overhead} = 8 \\
 \text{freepage} &= 0 \\
 \text{usable page size} &= (4096 - 22) * ((100 - 0) / 100) = 4074 \\
 \text{records per page} &= \text{FLOOR}(4074 / (4047 + 8)) = 1
 \end{aligned}$$

A single data page can carry one (1) recovery/restart entry. We will start off by building a 10 entry table (the "number of rows") in calculating space. Also note that when "freepage" is zero (0), the total pages is equal to the pages used.

$$\begin{aligned} \text{pages used} &= 2 + \text{CEILING}(10 / 1) = 12 \\ \text{total pages} &= \text{FLOOR}(12 * ((1 + 0) / 0)) = 12 \\ \text{est. kilobytes} &= 12 * 4 = 48 \end{aligned}$$

Though an index is really not required for this simple data space, the following calculation illustrates the size required if used.

$$\begin{aligned} \text{available space} &= (100 - \text{pctfree}) / 100 \\ \text{entries per leaf page} &= \text{FLOOR}(\text{available space} * 4050 \\ &\quad / (\text{key length} + 4)) \\ \text{entries per non-leaf page} &= \text{FLOOR}((\text{available space} * 4050) - 3) \\ &\quad / (\text{key length} + 3)) + 1 \\ \text{number of leaf pages} &= \text{CEILING}(\text{number of table rows} \\ &\quad / \text{entries per leaf}) \\ \text{number of non-leaf pages} &= \text{CEILING}(\text{number of leaf pages} \\ &\quad / \text{entries per non-leaf page}) \\ \text{total index pages} &= \text{number of leaf pages} \\ &\quad + \text{number of non-leaf pages} \\ \text{est. kilobytes} &= 4 * (\text{total index pages} + 2) \\ \text{where: pctfree} &= 0 \quad \text{key length} = 24 \\ \text{available space} &= (100 - 0) / 100 = 1 \\ \text{entries per leaf page} &= \text{FLOOR}(1 * 4050 / (24 + 4)) = 144 \\ \text{entries per non-leaf page} &= \text{FLOOR}((1 * 4050) - 3) / (24 + 3)) + 1 \\ &= 150 \\ \text{number of leaf pages} &= \text{CEILING}(10 / 144) = 1 \\ \text{number of non-leaf pages} &= \text{CEILING}(1 / 150) = 0 \\ &\quad \text{None used when leaf page} = 1 \\ \text{total index pages} &= 1 + 0 = 1 \\ \text{est. kilobytes} &= 4 * (1 + 2) = 12 \end{aligned}$$

This estimated index size will allow for an approximate 144 recovery/restart entries.

The following table reflects the aforementioned DB2 options required to build AIS+ AISRCOV.

<b>Tablespace DDL Option</b>	<b>Value</b>	<b>Description</b>
SEGSIZE	N/A	Simple tablespace does not use segments.
NUMPARTS	N/A	Simple tablespace does not use partitions.
LOCKSIZE	PAGE or ANY	The table is frequently updated where only 1 row fits to a page.
PRIQTY	48	See calculation above.
SECQTY	0	None required if primary quantity is set to carry all defined entries.
PCTFREE	0	The table is regularly updated, but space is not needed.
FREEPAGE	0	The table is regularly updated, but space is not needed.
BUFFERPOOL	BP0	Most shops use bufferpool 0 for the majority of their work.
CLOSE	NO	Table is constantly being accessed.
ERASE	NO	The data in the table does not need to be erased whenever the table is dropped.
<b>Index DDL Option</b>	<b>Value</b>	<b>Description</b>
UNIQUE		Index is unique. RCOV_PROG_NAME (ASC) RCOV_STEP_NAME (ASC) RCOV_JOB_NAME (ASC)
SUBPAGES	1	Table is small enough where subpages only needs to be 1.
PRIQTY	12	See calculation above.
SECQTY	0	None required if primary quantity is set to carry all defined entries.
PCTFREE	0	The table is regularly updated, but space is not needed.
FREEPAGE	0	The table is regularly updated, but space is not needed.
BUFFERPOOL	BP0	Most shops use bufferpool 0 for the majority of their work.

Index DDL Option	Value	Description
CLOSE	NO	Table is constantly being accessed.
ERASE	NO	The data in the table does not need to be erased whenever the table is dropped.

The following information describes the contents of an AIS+ AISRCOV data row:

Column	Type & Length	Key	Description
RCOV_PROG_NAME	CHAR (8) NNWD	1:1A	Name of program requiring recovery/restart entry.
RCOV_STEP_NAME	CHAR (8) NNWD	1:2A	Name of JCL step requiring recovery/restart entry.
RCOV_JOB_NAME	CHAR (8) NNWD	1:3A	Name of job requiring recovery/restart entry.
RCOV_CNTL_FREQ	CHAR (4) NNWD		
RCOV_CNTL_INTERNAL	CHAR (56) NNWD		Name of Batch Run Id taken from the CNTL file
RCOV_JOB_STATUS_CD	CHAR (1) NNWD		
RCOV_JOB_TERM_FL	CHAR (1) NNWD		
RCOV_EFF_PEAK_TIME	TIME NNWD		Time of day that is classified as peak utilization, where resource utilization is a concern.
RCOV_PEAK_CMT_ROWS	SMALLINT NNWD		The number of rows that will be processed before a commit during peak time.
RCOV_PEAK_CMT_SECS	SMALLINT NNWD		
RCOV_EFF_BTCH_TIME	TIME NNWD		Time of day that is not classified as peak utilization, where resource utilization is not a major concern.
RCOV_BAT_CMT_ROWS	SMALLINT NNWD		The number of rows that will be processed before a commit during non-peak time.
RCOV_BAT_CMT_SECS	SMALLINT NNWD		
RCOV_EXCL_LOCK_FL	CHAR (1) NNWD		
RCOV_START_TIME	TIME NNWD		

Column	Type & Length	Key	Description
RCOV_COMMIT_TIME	TIME NNWD		
RCOV_COMMIT_COUNT	SMALLINT NNWD		
RCOV_RUN_NUMBER	SMALLINT NNWD		
RCOV_MAX_RTRY_911	SMALLINT NNWD		
RCOV_MAX_RTRY_RST	SMALLINT NNWD		
RCOV_INP_FILE_CNT	INTEGER NNWD		The number of images stored during the run
RCOV_LOW_KEY	VARCHAR (3) NNWD		The Record Id for the last successfully stored image.  Only used for informational purposes
RCOV_HIGH_KEY	VARCHAR (3) NNWD		The record number at which processing should begin when a recovery scenario is encountered.
RCOV_RUN_INFO	VARCHAR (3862) NNWD		Basic information pertaining to the run.

## Appendix B: Storage of Image Overlays

### Feature Set-Up

This appendix introduces the VSR Image Overlay Storage feature, which allows storage of externally generated forms overlays in VALIN format into OAM. No updates to the front end application tables are made. Job AIFP01P accomplishes this task executing module AIFP001P in a similar manner to the normal VSR image storage feature. Perform the following steps to enable this feature of the product:

1. Modify and use the provided sample JCL, VSR30.JCLLIB(LINKFP01), to linkedit AIFP001P. Verify that the linkedit step links the modules into an AFP authorized library and that DSNALI is referenced via an include statement.

```
//????????? JOB (????,????), '@@@@@@',
//          CLASS=?,MSGCLASS=?,NOTIFY=??????
//*
//JOBLIB DD DSN=SYS1.V2R3.DSNLOAD,DISP=SHR
//*
//*****
//*
//          SAMPLE JOB TO LINKEDIT PROGRAM AIFP001P
//*
//          ADD YOUR JOBCARD
//*
//          MODIFY THE OVERRIDES ON THE PROC STATEMENT TO MATCH YOUR
//          DEFINED LIBRARIES
//*****
//          *****
//LNKBTCH PROC MODULE='PROGRAM',
//          REG=1024K,
//          LKEDPARM='XREF,LET,LIST,MAP,AMODE(31),RMODE(ANY)',
//          COB2LIB='SYS1.COB2LIB',
//          DSNLOAD='SYS1.V2R3.SDSNLOAD',
//          DSNEXIT='SYS1.V2R3.SDSNEXIT',
//          CSSLIB='SYS1.CSSLIB',
//          EYPLMD1='SYS1.SEYPLMD1',
//          LOADLIB='AIS.VSRV30.LOADLIB',
//          AISHLQ='AIS.VSRV30',
//          OUTC='*'
//          *****
//LKED EXEC PGM=IEWL,REGION=&REG,PARM='&LKEDPARM'
//SYSLIB DD DSN=&COB2LIB,DISP=SHR
//          DD DSN=&DSNLOAD,DISP=SHR
//          DD DSN=&DSNEXIT,DISP=SHR
//          DD DSN=&CSSLIB,DISP=SHR
//          DD DSN=&EYPLMD1,DISP=SHR
//          DD DSN=&LOADLIB,DISP=SHR
//SYSLIN DD DSN=&AISHLQ..OBJLIB(&MODULE),DISP=SHR
//          DD DDNAME=SYSIN
//SYSLMOD DD DSN=&LOADLIB(&MODULE),DISP=SHR
//SYSUT1 DD SPACE=(1024,(50,50)),UNIT=SYSDA
//SYSPRINT DD SYSOUT=&OUTC
// PEND
//          *****
//          LINKEDIT FOR AIFP001P
//          *****
//LINKFP1P EXEC LNKBTCH,
//          MODULE=AIFP001P
//LKED.SYSIN DD *
//          INCLUDE SYSLIB(DSNALI)
//          *****
//          *****
```



2. Modify JCL VSR30.JCL(BINDFP01) to conform to your sites standards and to include the DBRM modules required to support your storage groups (CBRISS00 for storage groups 00 through 04, CBRISS05 for storage groups 05 through 09, etc.). Submit this job to bind the VSR DB2 plan.

```
//????????? JOB (????,????), '@@@@@@', CLASS=?,MSGCLASS=?,NOTIFY=???????,00001000
//          REGION=1024K                                00002000
//*                                                00003000
//*****
//***** CHANGE ALL ? AND @ ABOVE *
//***** CHANGE ALL &&OAMDBRM TO YOUR SYSTEM DBRM LIBRARY THAT *
//***** CONTAINS THE CBRNNNNNN MEMBERS *
//***** CHANGE ALL &&HIQUAL TO YOUR VSR HIGH LEVEL QUALIFIER *
//***** CHANGE ALL &&HIQUAL2 TO YOUR AIS+ HIGH LEVEL QUALIFIER *
//*****
//*                                                00030000
//JOB LIB DD DSN=AISDB2.DSNLOAD,DISP=SHR                00040000
//BINDSTEP EXEC PGM=IKJEFT01,DYNAMNBR=20              00050000
//DBRMLIB DD DSN=&&OAMDBRM.DBRMLIB,DISP=SHR            00060000
//          DD DSN=&&HIQUAL.DBRMLIB,DISP=SHR            00080000
//SYSTSPRT DD SYSOUT=*                                00090000
//SYSPRINT DD SYSOUT=*                                00100000
//SYSUDUMP DD SYSOUT=*                                00110000
//SYSTSIN DD *
DSN SYSTEM(DB2S)
BIND PLAN(AIFP001P) -
MEM(AIFP001P,CBRHTBSV,CBRISS00,CBRISS05) -
ACT (REP) -
OWNER (AISADM) -
RETAIN -
ISOLATION (CS) -
VALIDATE (BIND) -
ACQUIRE (USE) -
RELEASE (COMMIT) -
EXPLAIN (NO)
END
/*
//*
//
```

3. Insert a row into the AIS+ recovery table for default Forms Overlay run parameters. These parameters will stay stagnant for every run of the VSR process. The sample SPUFI member INSRCOV in the VSR JCL library should be used to insert the row. If you are not running AIS+, please refer to Appendix A of this manual for information concerning the creation of the recovery table.

## Running the Forms Overlay Storage Feature

Records input to the VSR Image Overlay Storage process can be from ODM V1.2, V2.1.1, or V2.2. Control file options must be set to indicate the type of records being input to the process. VSR reformats the input data into ODM V2.2 record format, which is used in the processing of records internally. Objects are stored in ODM version 2 format.

The VSR Image Overlay Storage control file, CNTLFP01, contains control options configuring the processing. This information is input into the process and verified as the initial step in the process. Any errors in the control file halt the processing, and require correction and resubmittal of the job.

Processing options entered should be carefully considered. The RUNCNTRL variable enables the user to select the processing flow. The control file in conjunction with a recovery entry can be used to restart the process in the event of a job execution interruption. The following details the fields entered in the file, and provides a sample input file format.

<b>VSR Control File CNTLFP01</b>				
<b>Keyword</b>	<b>Key Number</b>	<b>Size of Value (Bytes)</b>	<b>Req</b>	<b>Description</b>
BATRUNID=	01	Char (20)	O	The batch run ID is a 20-character name given by the user to identify the run name. The default is the name generated by the validation process in the form of a DB2 datestamp. This ID, if unique, can be used in the restarting of the process in the event of a failure.
RUNCNTRL=	02	Num (2)	R	The run control option has the following values: 01 - Normal run, all records stored 02 - Validate control file only 03 - Print only, all records 04 - Print only, all records, validate and print errors 05 - Normal run, all records, store, record range 06 - Print only, record range
STRECORD=	03	Num (9)	O	The start record ID is used when the run control option is 05 or 06. It represents the beginning record in a processing range, and must be an index record. This is equivalent to the Record ID number in the input data sets. This value is overridden in a recovery scenario by the record value of the last processed record in the recovery table AISRCOV. For ODM V1.2 and V2.1 this field must be six characters only, zero filled. For ODM V2.2 this field is nine positions, zero filled if necessary.
ENRECORD=	04	Num (9)	O	The end record ID is used when the run control option is 05 or 06. It represents the last record in a processing range, and must be an index type record. This is equivalent to the Record ID number in the input data sets. For ODM V1.2 and V2.1 this field must be six characters only, zero filled. For ODM V2.2 this field is nine positions, zero filled if necessary. For range processing either ENRECORD or NMOBJECT must be used, with only one value allowed.

<b>VSR Control File CNTLFP01</b>				
<b>Keyword</b>	<b>Key Number</b>	<b>Size of Value (Bytes)</b>	<b>Req</b>	<b>Description</b>
NMOBJECT=	05	Num (9)	O	The number of objects is used when the run control option is 05 or 06. It is used instead of the end record to request the number of objects to process, starting at the record number specified in the start record. For range processing either ENRECORD or NMOBJECT must be used, with only one value allowed.
MESGDEST=	06	Num (2)	O	The message destination code identifies the mode of display for error and progress messages. The values are: 01 - All messages are sent to the system output (SYSOUT) destination. The default is 01. 02 - All messages are sent to the SYSOUT destination and to the console.
MSGCOUNT=	07	Num (3)	O	The progress message count generates a progress message after the number of index records specified are processed. If this value is 0, which is the default, then no progress messages are displayed or printed. The message count is 0 to 999.
ERRTHRES=	08	Num (4)	O	The error threshold value is the number of errors tolerated. Each error encountered in validation of fields increments the error count by one. It is in the range of 0 to 9999. If this value is 0, which is the default, the error threshold processing is turned off.
PRNTPAGE=	09	Num (2)	O	The print page value is in the range of 1 to 99. It represents the number of lines specified to be printed on one page, not including header lines. The default value is 60.
VERSRELS=	10	Char (6)	R	The version and release name value must be 020100 for V2.1, 0201.1 for V2.1.1, or 020200 for V2.2.
ODMCNTRL=	11	Num (2)	R	For FAF: The IODM control ID record value must be 01 or 02. The values are: 01 - Use IODM ID from ODMRECID 02 - Use IODM ID from the input file.
ODMRECID=	12	Char (4)	O	For FAF: The IODM ID identifies where the object is stored. It is required if the value of ODMCNTRL is 01.
DB2SYSID=	13	Char (4)	R	The DB2 subsystem ID identifies where the VSR process runs. VSR currently does not support multiple DB2 subsystems for OAM.
USERINFO=	14	Char (54)	O	The user reference information contains the title information that the user wants to appear in reports.

<b>VSR Control File CNTLFP01</b>				
<b>Keyword</b>	<b>Key Number</b>	<b>Size of Value (Bytes)</b>	<b>Req</b>	<b>Description</b>
STRVRELS=	15	Char (6)	R	The version and release of the input file. A value of 0201 1 means that a V2.1 input file is used with a block size of 32,760 bytes. A value of 0102 1 means that a V1.2 input file is used with a block size of 32,754 bytes. A value of 0202, which is the default, means that a V2.2 input file is used with a block size of 32760. The value must be set appropriately based on desired input record format.
V12DATEF=	16	Num (2)	O	The V1.2 date format is required if a V1.2 input file is used. Valid values are from 01 to 28
KEEPRCOV=	17	Char (01)	O	Flag to determine if Recovery table entry for run should be kept at successful completion of a process run. Valid values are "Y", keep the table entry, or "N" delete table entry at the end of the processing run. A "C" is placed in the Recovery table RCOV_JOB_TERM_FL field for successful program executions with return codes < +9.
COLLNAME=	18	Char (44)	R	Specifies the collection to store the forms overlay in.

**Note:**

Column positions:

**Keyword** - 1 to 9

**Key Number** - 25 to 26 (optional)

**Value** - starts in column 27.

Comment records may be imbedded in the control parameter file by specifying an asterisk (\*) in position 1.

The key number begins in column position 25 because the first 24 columns can be used to hold the keyword.

In the Size of Value column, the abbreviations are as follows:

**Char** – Alphanumeric

**Num** - Numeric.

In the REQ column, the abbreviations are as follows:

**R** – Required

**O** - Optional. The optional fields remain optional unless a dependent field makes is required. For example, ODMRECID record is required if the value of ODMCNTRL is 01.

A sample CNTRLDAT file follows:

```
* TEST CONTROL FILE FOR BATCH STORE PROCESS.
BATRUNID=          01TESTFORM RUN
RUNCNTRL=          0201
STRECORD=          03000000001
ENRECORD=          04
NMOBJECT=          05-00000005
MESGDEST=          0701
MSGCOUNT=         08005
ERRTHRES=          090005
PRNTPAGE=          1060
VERSRELS=          11020200
ODMCNTRL=          1202
ODMRECID=          13
DB2SYSID=          14DB2
USERINFO=          15TEST FORMS STORE TITLE
STRVRELS=          160102 1
V12DATEF=          1001
TIMEFRMT=          221
DATEFRMT=          234
KEEPRCOV=          32Y

          COLLNAME=          34EYP.COLLECTO
```

See section VSR Input File Layouts for a description of each input file format allowed. The only validation of the input file occurs for the object type or class, which should be 'J' or hex value '8020' respectively. Any other type objects in the file will be rejected and a return code of '04' will be returned. This return code is valid for this process and should be ignored.

The following is sample execution JCL. Please update for your site specific standards

```
//????????? JOB (????,????), '@@@@@@', CLASS=?,MSGCLASS=?,          00001000
//          NOTIFY=?????????          00001000
//*          00003000
//*****
//***** CHANGE ALL ? AND @ ABOVE *
//***** CHANGE ALL &&DB2SYS BELOW TO YOUR INSTALLATIONS SYBSYSTEM ID*
//***** CHANGE ALL &&DB2.DSNLOAD TO YOUR DB2 SYSTEM LOADLIB *
//***** CHANGE ALL &&SITEHLQ TO YOUR VSR HIGH LEVEL QUALIFIER *
//***** CHANGE ALL &&MSGCLASS TO YOUR DESIRED OUTPUT CLASS OR '*' *
//***** CHANGE ALL &&UNIT TO YOUR INSTALLATIONS DASD UNIT *
//***** CHANGE ALL &&HSCS TO YOUR HSCS FILE NAME EXTENSIONS *
//*****
//*
//VSRPROC PROC HIQUAL='&&SITEHLQ',
//          MSGOUT='&&MSGCLASS'
//          UNIT='&&UNIT'
//          HSCS='&&HSCS'
//*
//STEP001 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=&MSGOUT
//ERRRLOG1 DD DSN=&HIQUAL..FP01PROD.ERLOG1,
//          DISP=(MOD,DELETE,DELETE),UNIT=&UNIT,SPACE=(TRK,0)
//WORKRPT1 DD DSN=&HIQUAL..FP01PROD.WRKRPT1,
//          DISP=(MOD,DELETE,DELETE),UNIT=&UNIT,SPACE=(TRK,0)
//OUTDATA DD DSN=&HIQUAL..FP01PROD.OUTDAT,
//          DISP=(MOD,DELETE,DELETE),UNIT=&UNIT,SPACE=(TRK,0)
//ERRDATA DD DSN=&HIQUAL..FP01PROD.ERRDAT,
//          DISP=(MOD,DELETE,DELETE),UNIT=&UNIT,SPACE=(TRK,0)
//*
//*
//STEP002 EXEC PGM=AIFP001P,COND=(0,NE),DYNAMNBR=30,
//          TIME=1440,REGION=30M
//STEPLIB DD DSN=&&DB2.DSNLOAD,DISP=SHR
//SYSOUT DD SYSOUT=&MSGOUT
//SYSABOUT DD SYSOUT=&MSGOUT
```

```

//SYSUDUMP DD SYSOUT=&MSGOUT
//SYSTSPRT DD SYSOUT=&MSGOUT
//SYSPRINT DD SYSOUT=&MSGOUT
//ERRRLOG1 DD DSN=&HIQUAL..FP01PROD.ERLOG1,DISP=(,CATLG,KEEP),
//          UNIT=&UNIT,SPACE=(133,(25,10),RLSE),
//          DCB=(RECFM=FB,LRECL=133)
//WORKRPT1 DD DSN=&HIQUAL..FP01PROD.WRKRPT1,DISP=(,CATLG,KEEP),
//          UNIT=&UNIT,SPACE=(133,(25,10),RLSE),
//          DCB=(RECFM=FB,LRECL=133)
//OUTDATA  DD DSN=&HIQUAL..FP01PROD.OUTDAT,DISP=(NEW,CATLG,KEEP),
//          UNIT=&UNIT,AVGREC=K,SPACE=(CYL,(5,5),RLSE),
//          DCB=(RECFM=VB,LRECL=32750,BLKSIZE=32754)
//ERRDATA  DD DSN=&HIQUAL..FP01PROD.ERRDAT,DISP=(NEW,CATLG,KEEP),
//          UNIT=&UNIT,AVGREC=K,SPACE=(CYL,(25,20),RLSE),
//          DCB=(RECFM=VB,LRECL=32750,BLKSIZE=32754)
//CNTRLDAT DD DSN=&HIQUAL..VIPSVSR.PARMLIB(CNTLFP01),DISP=SHR
//INDATA1  DD DSN=&HIQUAL..&HSCS,DISP=SHR,
//          DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760)
//VSRPROC  PEND /***** END OF PROC VSRPROC *****/
//*
//*****
//VSRRUN   EXEC VSRPROC
//
//*
//*****
//* IF YOU ARE RESTARTING DUE TO ERRORS, PLACE THE PROPER INFORMATION *
//* IN THE PARMS BELOW AND REPLACE THE EXEC STATEMENT ON STEP002 *
//* BBBBBBBBBBBBBBBBBBBB = THE BATCH RUN ID OF THE ORIGINAL RUN FROM *
//*                       THE CONTROL FILE *
//* DDDDDDDD             = A RESTART DATE - THIS VALUE IS FOUND IN THE *
//*                       RECOVERY TABLE IN COLUMN RCOV_JOB_NAME *
//*                       FORMAT IS YYYYMMDD *
//* TTTTTTTT           = A RESTART TIME - THIS VALUE IS FOUND IN THE *
//*                       RECOVERY TABLE IN COLUMN RCOV_STEP_NAME *
//*                       FORMAT IS HH:MM:SS *
//*****
//STEP002  EXEC PGM=AIFP001P,COND=(0,NE),DYNAMNBR=30,
//          TIME=1440,REGION=30M,
//          PARM='BBBBBBBBBBBBBBBBBBBB DDDDDDD TTTTTTT'

```