CICS VSAM Transparency for z/OS
Version 2 Release 1

User's Guide

SC34-7249-00

CICS VSAM Transparency for z/OS
Version 2 Release 1

User's Guide

SC34-7249-00

Note
Before using this information and the product it supports, read the information in "Notices" on page 177.

This edition applies toVersion 2 Release 1 of the CICS VSAM Transparency for z/OS, program number 5655-Y01, and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Contents

Preface

This book explains how to use CICS® VSAM Transparency (CICS VT) as a first step to exploiting DB2®. It contains detailed steps on how to use data migration and application transparency and puts them in the context of the overall conversion process. It also provides extensive information to help you understand how you can gain maximum benefit from using CICS VT, and advice on the optimal design of your DB2 objects.

Unless you are familiar with CICS VT and are using the book for reference only, you should read each chapter in turn as the information builds on that contained in previous chapters.

Who this book is for

This book is primarily aimed at application programmers and database administrators, but can also be used by managers and those who are responsible for planning and performing a database conversion.

What you need to know to understand this book

To use the information in this book, you need to have a basic knowledge of VSAM, DB2, and application programming. Familiarity with z/OS®, CICS, and JCL is also desirable.

How to use this book

This book is split into sections that contain information on installing and customizing CICS VT, and data migration. Each section is self-contained. Use an individual section as a guide when performing the tasks described in it.

Other product publications

The CICS VT documentation includes one other book:

CICS VT Data Reengineering and Customization Guide.

To use CICS VT effectively requires knowledge and experience of DB2 and CICS, and the following IBM books contain information that the CICS VT user will find helpful.

DB2 SQL Reference
DB2 Utility Reference
CICS System Definition Guide
CICS DB2 Guide
CICS Performance Guide

Notes on terminology

In this book, the term CICS, used without any qualification, refers to the CICS element of IBM® CICS Transaction Server for z/OS. The term DB2 refers to DB2 UDB for z/OS and OS/390®.

Conventions used in this book

The following conventions are used throughout this book:

Bold text is used for:
Emphasis of key parameters
Italic text is used for:
Variable names
`Monospace` text is used for:
JCL statements
SYSIN control cards
DB2 SQL
Code or syntax examples

Summary of changes

This section summarizes what is new and what has changed in CICS VSAM Transparency for z/OS version 2.1 under the following headings:

- Mapping and data migration
- Batch processing set up
- REPRO support
- National language support
- Tracing

Mapping and data migration

Enhancements to COBOL binary fields support are introduced in version 2.1, including the use of the BIGINT DB2 column data type for doubleword binary fields. Support for COMP-5 fields is introduced.

There are new warning messages for unsigned COBOL COMP and COMP-5 fields to highlight potential data conversion issues.

There is a new facility to generate the appropriate batch job JCL and control cards for data migration. This is described in "Automated data migration" on page 81.

Batch processing setup

The JCL changes that are required for CICS VT can be made prior to migrating the VSAM file. A mechanism is introduced to simplify switching between VSAM and CICS VT DB2 access. This is described in "Making JCL changes before migration" on page 99.

REPRO support

In previous versions of CICS VT, support for REPRO was restricted to read only. Version 2.1 supports REPRO using the REPLACE and REUSE options. This is explained in "CICS VT and IDCAMS" on page 131.

National language support

All ISPF screens in version 2.1 are written in dialog tag reference (DTL) to simplify national language translation.

Tracing

Version 2.1 allows CICS VT tracing in CICS at the DIM level. You no longer have to supply one or more specific TERMIDs. This is explained in "Activating the trace in CICS" on page 139 .

Chapter 1. Introduction to CICS VT

VSAM was first released with the IBM MVS™ operating systems OS/VS1 and OS/VS2 in the early 1970s. It has been the underlying access method for the IBM DB2 software on MVS and OS/390 since the first release. It is also the most common access method for IMS™ databases.

As well as being the core of IBM relational and hierarchical database management systems, VSAM became the preferred choice for CICS users when support was introduced in 1974.

The IBM z/OS Database Management System (DBMS) provides an operability layer that interfaces with the data access layer. Some of the main functions provided by the DBMS layer are:

- Data integrity
- Data sharing
- Standard API
- Availability
- Scalability

These functions are limited for programs that access VSAM files natively.

A statement of the current problem

It is essential to understand the nature of the problem before you can evaluate different solutions. Within the context of CICS VT there are several issues with VSAM, the main ones being:

- It is difficult to share the data across different processes and maintain data integrity.
- There are architectural limits that impose a maximum size on a VSAM file.
- Ad hoc access to VSAM files by knowledge workers or decision support users is difficult. This impacts the ability of your end users to support rapidly changing business trends.
- There are no data mining tools for VSAM.

There are many applications for which these issues do not apply, and these applications might not be candidates for CICS VT. However, other applications are disadvantaged by some or all of these factors. CICS VT is one of a number of alternative migration options.

Possible solutions

A number of solutions exist, but the three common ones follow.

Option 1 - convert your VSAM files to DB2

Assuming that you have legacy applications accessing VSAM files natively, you are faced with two major activities:

- The initial conversion of the data from VSAM to DB2.
- The changes to your application programs to access the DB2 data.

This is a high-risk, high-cost strategy.

Option 2 - extract VSAM data to DB2

With this approach, your risks and costs are reduced. The problem is that your DB2 data is not current. Also, unless you enforce a read-only rule in DB2, you have issues with data synchronization.

Another issue is that as business changes, your requirements change and your ability to be responsive to these changes is impinged by this strategy.

Option 3 - duplicate the VSAM files in DB2

This approach eliminates some of the restrictions in the previous approach, but you still have the potential issues of synchronization. You have to invest a lot of time and effort into writing interface programs, and your disaster recovery procedures can become very complicated.

The CICS VT solution

Uniquely, the CICS VT solution means that you do not have to change any of your existing application programs, but your data is still migrated to DB2. Because you don't change your programs, the migration costs are reduced, but more importantly your risk is substantially reduced.

By using CICS VT, you have a single copy of the data in DB2 without having to change any of your existing legacy application programs. This means that the inherent problems associated with the previously described options are eliminated, as follows:

- Only one copy of the data exists, so synchronization is not an issue. Your legacy programs and your end users can share the data.
- No changes to legacy programs are required. The degree of change and therefore the scale of the risk is reduced significantly.
- Local and disaster recovery procedures are simplified because there is one less data type to support.
- You can use standard tools, skills, and resources, and achieve economies of scale in your database administration area.
- Your data integrity is assured because it is under the control of a DBMS.

As well as these benefits, your end users can have real-time access to real-time data.

Data reengineering with CICS VT

The primary purpose of CICS VT is to provide a transparent interface to your legacy programs, so that data can be retrieved from DB2 without any program changes. At the same time, you can enhance the value of your existing data by using the data re-engineering capabilities provided by CICS VT.

For example, the date fields in your VSAM file can become DB2 DATE columns, and CICS VT will handle the data translation automatically. Similarly, zoned decimal fields can become integer or decimal columns in DB2 . This allows you to exploit many of the built-in SQL functions when accessing the DB2 tables outside of your legacy programs.

Further data re-engineering with CICS VT API

In addition to the re-engineering capabilities that CICS VT invokes automatically, two APIs are provided for more complicated restructuring. For example, assume that you have multiple copybooks for the same file and apart from the key, the copybook attributes are completely different. Also assume that your users want the file separated into multiple DB2 tables, according to the copybook type.

This DB2 implementation is possible using a combination of two types of CICS VT user exits. These are:

- A field level exit to perform re-engineering of one or more fields. This is known as a field build exit (FBE).
- A record level exit that performs re-engineering of an entire record. This exit is triggered by a call that inserts a new record, replaces an existing record, or deletes an existing record. This is known as an insert, replace, delete (IRD) exit.

In the current example where an existing file needs to be split into separate DB2 tables, these exits would operate as follows:

FBE This exit would decide which table contains the required records, by analyzing a specific field or key value in the record. It contains the necessary SQL to process retrieval calls.

IRD This exit would handle update calls, and contains the necessary SQL to update the DB2 tables.

The coding of user exits is described in the manual *CICS VT Data Reengineering and Customisation Guide* .

CICS VT positions you for the future

In application development, there are two areas where CICS VT can provide significant advantages:

- Your existing legacy applications can be maintained as SQL applications. You can add SQL statements to these programs and let CICS VT continue to handle your VSAM calls. This can extend the life cycle of your legacy applications, by making it easier to maintain and enhance them.
- You can write new application programs to access the new DB2 tables, and leave DB2 to manage concurrent access and data integrity. CICS VT does not restrict your ability to do this.

Features not supported by CICS VT

The following features are not supported by this release of CICS VT:

- Direct access to a VSAM alternate index cluster as if it had been defined as a VSAM base cluster.
- ESDS data set types.
- Linear data sets.
- Processing using RBA access.

General considerations

A key feature of CICS VT is that your application programs are unchanged although your VSAM data is migrated to DB2. In practice, there may be situations when some minor program changes overcome operational or performance issues. Typical situations that may benefit from minor application changes are:

- Long running CICS transactions that do not issue SYNCPOINTs on a regular basis may cause DB2 deadlocks or timeouts. Row-level locking may help, but it may be necessary to add SYNCPOINT calls to offending application programs.
- Batch programs that use REPRO, SORT and SYNCSORT work with CICS VT, but it may make more sense to use equivalent DB2 utilities to achieve optimal performance, especially for large files.
- Batch programs that issue many UPDATE commands may cause contention issues with other DB2 tasks, and may exceed NUMLKTS and NUMLKUS limits that are defined in your DSNZPARM. One solution may be to add SQL COMMIT calls to your application program. This is discussed in "Adding SQL commit calls" on page 130

If you are aware of any long running CICS transactions, or are migrating large files that are updated heavily by batch programs, you should include the possibility of minor application program changes in your migration planning.

Chapter 2. Configuration and setup

When the SMP/E installation is complete, you are ready to configure CICS VT for your environment. If you are upgrading from version 1.2 to version 2.1, please go to "Upgrading from CICS VT version 1.2" on page 21.

A summary of the configuration and setup tasks follows:

1. "Running the installation customization application" on page 6
2. "Creating the CICS VT system objects in DB2" on page 11
3. "Creating CICS VT user data sets" on page 12
4. "Installing the batch subsystem" on page 12
5. "Installing CICS VT in CICS" on page 13
6. "Additional configuration actions" on page 15
7. "Performing the installation verification procedure (IVP)" on page 17

Different skills and system authorities are required at different points in the installation process. The following table indicates the suggested skills for each of the previous steps:

Role	Actions
DB2 administrator	Perform steps 1, 2, 3, and 7
MVS system programmer	Perform steps 4 and 6
CICS system programmer	Perform step 5

Input from application developers who are involved in the CICS VT migration project is desirable in steps 1 and 7.

In this documentation, data sets prefixed by *my* refer to the customised libraries that are created during the tailoring program. Read access to these data sets is required by personnel who perform VSAM file migrations.

Data sets prefixed by *appl* are created by one of the installation batch jobs. Update access to these data sets is required by personnel who perform VSAM file migrations.

National language support

There are three CICS VT data sets that are enabled for national language support:

- VID.SVIDMLI*s*
- VID.SVIDPLI*s*
- VID.SVIDLOD*s*

Each data set has a single character suffix *s* to indicate the national language specified during configuration. The languages supported by version 2.1 are English ('E') and Japanese ('K' for Kanji). Note that references to data set names in sample jobs use the English data set names.

Running the installation customization application

The configuration of the CICS VT libraries is performed by an ISPF application. It creates two new libraries:

- *my*.SVIDCNFG.*custom*
- *my*.SVIDSAMP.*custom*

where *my* and *custom* represent any qualifiers you choose.

All of the jobs that are required to perform and test the CICS VT installation are in *my* .SVIDCNFG. *custom* . CICS VT utilities, sample CICS VT jobs, sample DB2 utilities, and sample CICS VT exits are in *my* .SVIDSAMP. *custom* .

In general, the configuration parameters that you specify relate to the system where the data set mapping and data migration is performed. New data sets that are created by installation jobs are used by application programmers and DBAs. Input from both areas is desirable prior to starting the configuration process, and this is especially important if you are running it on a systems programmer machine or sandbox.

The tailoring program

To start the tailoring application using default parameters, run the following TSO command:

```
EXEC 'cvt210.SVIDEXEC(VIDSETUP)'
```

cvt210 is the high-level qualifier of your SMP/E data sets. The default parameters for VIDSETUP assume that the high-level qualifier of the *cvt210* .SVIDEXEC library is the same for all of the CICS VT SMP/E data sets, and you are using the default national language English. If you move the VIDSETUP member to a common ISPF library, called SYS3.EXEC for example, and you are using the Kanji language, you can override the default parameters as follows:

```
EXEC 'SYS3.EXEC(VIDSETUP)' 'hlq(cvt210) lang(k)'
```

The REXX module VIDINST performs the configuration. The tailoring application has four ISPF panels and these are explained in the next section.

The member VID@READ in the output *my*.SVIDCNFG.*custom* library contains a list of the customization values you supply during configuration. You should keep this member for documentation purposes. If the tailoring application encounters any errors, messages are written to this member.

You must supply a value for every field.

Tailoring panel 1 of 4

When you run VIDSETUP, the CICS VT installation tailoring application runs the customization program VIDINST. The first ISPF panel of the installation tailoring panels is shown in Figure 1 on page 7 .

```
-------------------------- CICS VT Setup (Screen 1/4) ------------------------
Command ===> ____________________________________________________________________
Insert data set names in the following screens. Do not use quotes.

CICS VT VIDCNFG library : cvt210.SVIDCNFG_____________________________

Custom VIDCNFG library  : my.SVIDCNFG.custom___________________________

CICS VT VIDSAMP library : cvt210.SVIDSAMP_____________________________

Custom VIDSAMP library  : my.SVIDSAMP.custom___________________________

MVS Assembler program   : ASMA90 (ASMA90/IEV90)

Output Device Type      : SYSDA__Generic device type (example: SYSDA)

Job card statement:
//VIDJOB JOB (ACCT,PGMer-name),CLASS=L,MSGCLASS=X,NOTIFY=&SYSUID
____________________________________________________________________________
____________________________________________________________________________
____________________________________________________________________________

Press Enter to continue, PF3 to Exit or PF1 for Help
```

Figure 1. CICS VT tailoring application panel 1

You should note the following:

- The data set names must be fully qualified.
- The CICS VT VIDCNFG and VIDSAMP data sets you specify are input data sets and are not updated during tailoring.
- The CUSTOM VIDCNFG and VIDSAMP data sets you specify are output data sets and are created during tailoring.
- You must specify whether your installation uses ASMA90 or IEV90 as the default MVS assembler program.
- You must specify a generic device type, such as SYSDA, for the output device type. All of the output data sets that are created during the configuration and setup will be created on this device type. You can move any of the data sets to a different volume after the configuration and setup is complete.

Optionally, you can define a skeleton job card that is added to all of the members of *my*.SVIDCNFG.*custom* and *my*.SVIDSAMP.*custom* that are MVS jobs.

The values that you supply in the remainder of the tailoring screens are recorded for reference in the VID@READ member in *my*.SVIDSAMP.*custom*.

Tailoring panel 2 of 4

Use the second tailoring panel to specify the DB2 parameter values, as shown in Figure 2 on page 8 .

```
------------------------- CICS VT Setup (Screen 2/4) -------------------------
 Command ===> _____________________________________________ Scroll ===> PAGE

 DB2 SDSNLOAD library  : DB2.SDSNLOAD_____________________________

 DB2 sample program    : DB2.RUNLIB.LOAD_____________________________
           library              (typically .RUNLIB.LOAD)

 DB2 subsystem         : DB2B____ (DB2 subsystem used for the CICS VT install)

 CICS VT qualifier     : CICSVT..................................... +
                              (DB2 qualifer for CICS VT system tables)

 CICS VT DB2 owner     : CICSVT__ (DB2 owner to bind CICS VT plans and packages)

 CICS VT DB2 database* : CICSVTDB (DB2 database for CICS VT system tables)

 Table storage group   : CICSVTSG (DB2 storage group for CICS VT system tables)


 * If database is new then uncomment 'CREATE DATABASE' SQL in member VIDIDB2

 Press Enter to continue, PF3 to go to previous panel or PF1 for Help
```

Figure 2. CICS VT tailoring panel 2 with sample values

The parameter value descriptions are as follows:

Table 1. User parameter descriptions for panel 2

Parameter	Description
DB2 SDSNLOAD library	The name of the DB2 load library (SDSNLOAD).
DB2 sample program library	The name of the library containing the compiled or assembled sample DB2 sample programs such as DSNTIAUL and DSNTEP2, usually called *db2hlq* .RUNLIB.LOAD. The sample program DSNTEP2 is used to create DB2 objects for automated mapping.
DB2 subsystem	The subsystem identifier of the DB2 subsystem where you map your VSAM data sets to CICS VT.
CICS VT qualifier	The DB2 qualifier that will be used as the CREATOR of the CICS VT control tables. This field is scrollable.
CICS VT DB2 owner	The DB2 authorization ID that will be used as the BIND owner for the CICS VT packages and plans.
CICS VT DB2 database	The DB2 database that will contain the CICS VT control tables. If you decide to create a new database, you will have to remove the comment from the SQL CREATE DATABASE statement in member VIDIDB2 in the customized SVIDCNFG library. Do this after you have run the customization dialogs.

Table 1. User parameter descriptions for panel 2 (continued)

Parameter	Description
Table storage group	The DB2 storage group where you will create the CICS VT control tables and indexes. If the storage group does not exist, manually tailor member VIDIDB2 in *my* .SVIDCNFG. *custom* after you run the customization dialogs. Instructions are included as comments in the VIDIDB2 member.

Tailoring panel 3 of 4

The next panel displayed by the ISPF install program is shown in Figure 3 .

```
-------------------------- CICS VT Setup (Screen 3/4) -------------------------
Command ===> ____________________________________________________________________

CICS VT load library   : cvt210.SVIDLODE_____________________________

CICS VT DBRM library   : cvt210.SVIDDBRM_____________________________

CICS VT ISPF qualifier : CICSVT______________________________________

New VT driver library  : appl.drivers.load___________________________

New VT driver DBRM     : appl.drivers.dbrm___________________________

CICS SDFHLOAD library  : CICS.SDFHLOAD_______________________________

CICS DFHCSD library    : CICS.DFHCSD_________________________________

CICS VT language suffix: E



Press Enter to continue, PF3 to go to previous panel or PF1 for Help
```

Figure 3. CICS VT tailoring panel 3 with sample values

The parameter value descriptions are as follows:

Table 2. User parameter descriptions for panel 3

Parameter	Description
CICS VT load library	The name of the SVIDLOD*s* distribution library, where *s* is the national language identifier.
CICS VT DBRM library	The name of the SVIDDBRM distribution library.
CICS VT ISPF qualifier	The prefix for the CICS VT ISPF data sets. For example, if the SVIDEXEC data set is called ACME.VID.V210.SVIDEXEC, the value for this parameter is ACME.VID.V120.

Table 2. User parameter descriptions for panel 3 (continued)

Parameter	Description
New VT driver library	The name of a new load library where you will generate the CICS VT application driver modules. Specify any data set name that conforms to your site standards. This library must be added to the CICS DFHRPL concatenation and to STEPLIB in batch JCL. See "Modifying the CICS startup JCL" on page 14 .
New VT driver DBRM	The name of a new DBRM library where you will generate the CICS VT DDM driver module DB2 DBRMs. Specify any data set name that conforms to your site standards.
CICS SDFHLOAD library	The name of your CICS SDFHLOAD library.
CICS DFHCSD library	The name of the CICS CSD data set where the CICS elements of CICS VT will be defined.
CICS VT language suffix	The single character national language suffix. The default is E representing English. You can also specify K for Kanji.

Tailoring panel 4 of 4

The final panel displayed by the ISPF tailoring application is a confirmation panel showing the names of the customized output data sets that will be created. Press Enter to continue, or PF3 to go back and review the panel values you have supplied.

```
------------------------- CICS VT Setup (Screen 4/4) -------------------------
Command ===> ____________________________________________________________

                              Confirmation screen
                              -------------------


The customized libraries are about to be created or modified.

   All members will be copied :

      from   cvt210.SVIDCNFG
        to   my.SVIDCNFG.custom

      from   cvt210.SVIDSAMP
        to   my.SVIDSAMP.custom

The original libraries will not be changed during the customization.




Press Enter to customize, PF3 to go to previous panel or PF1 for Help
```

Figure 4. CICS VT installation confirmation panel

A panel displays when the tailoring program has finished, showing the total number of members processed in each data set.

Review the VID@READ member in *my* .SVIDCNFG. *custom* . If the tailoring application encounters errors, appropriate messages are written to this member.

Rerunning the tailoring program

You can rerun the tailoring program if required. Parameter values are stored as ISPF variables and persist across each invocation of the tailoring program.

Creating the CICS VT system objects in DB2

CICS VT uses control tables in DB2 that are accessed using DB2 packages. You need appropriate DB2 privileges.

Ideally the CICS VT control tables should be created in their own database and update access restricted. The CICS VT application tables that will replace your migrated VSAM files should be in separate DB2 application databases.

Create DB2 table and index objects

This step creates control tables that are used for mapping VSAM files to DB2, and also the tables that CICS VT uses for the installation verification procedure (IVP).

The SQL data definition language (DDL) is in member VIDIDB2 in *my*.SVIDCNFG.*custom*. Review this member carefully and where necessary, manually edit it to conform to your local DB2 standards. The DDL includes DROP TABLESPACE statements as comments. These are provided to simplify rerunning the DDL.

Use SPUFI or another dynamic SQL program to run the SQL.

Managing the CICS VT control tables

The CICS VT control tables are used during mapping and for generating the CICS VT driver modules. When all of your VSAM data sets have been migrated to DB2 , the CICS VT control tables are no longer used unless you want to update mapping information.

Ensure that you have a policy in place to manage backup and recovery of these critical CICS VT components. You need to pay particular attention to the VID_SRC table. The data in this table is transient and is created and used when you generate CICS VT driver modules. After the drivers are generated, the data is no longer required.

DB2 object sizes

The size of the CICSVT DB2 objects created by member VIDIDB2 support the migration of approximately 200 data sets with 25,000 copybook fields. You might need to increase the size of one or more of the CICS VT page sets if your migration exceeds either of these figures.

Bind CICS VT packages and plans

Member VIDB2BND in *my*.SVIDCNFG.*custom* contains a job stream that binds the DB2 packages and plans needed by CICS VT system functions. No changes to the job stream should be required, although you must have the required DB2 privileges for the bind to be successful. Add your DB2 SDSNLOAD library to the STEPLIB DD in this job if the data set is not in the linklist.

CICS VT creates three collections in DB2. These collections are used as follows:

- Packages in the collection VIDSYS are used by the programs that generate the CICS VT driver programs.
- The CICS VT programs in the ISPF component are created in the package VIDISPF.
- Packages in the collection VIDCOLL are used by CICS VT to handle runtime errors.

The plans you use for your application programs must include VIDCOLL in the package list at SQL driver bind time. This is also the default collection for the application data set driver modules (DDMs), which are modules that CICS VT generates for each migrated data set. CICS VT also generates data set information driver modules (DIMs) for each migrated data set. DIMs and DDMs are covered in "Generating CICS VT runtime modules" on page 53.

The implications concerning the packages in VIDCOLL and your application drivers are discussed in "DB2 considerations" on page 58.

Creating CICS VT user data sets

Run the *my*.SVIDCNFG.*custom* member VIDDDEF to create the data sets for your application DIM and DDM drivers.

Installing the batch subsystem

CICS VT intercepts VSAM calls in your application batch programs using an MVS subsystem interface (SSI). The SSI can be installed dynamically if you are running OS/390 V2.4 or any later level. If you do not have OS/390 V2.4, you have to IPL to complete the SSI installation. Updates to your SYS1.PARMLIB data set are required to install the subsystem permanently.

Installing the subsystem, whether dynamically or permanently, does not mean that all VSAM requests are intercepted and processed by CICS VT. This only occurs when the appropriate JCL changes are made.

Dynamic SSI installation

To dynamically install a new MVS subsystem, load the initialization module from an existing linklisted library. To dynamically install the CICS VT SSI, perform the following steps:

1. Copy all the modules in VID.SVIDLINK to an existing linklisted data set.
2. Refresh the library lookaside (LLA).
3. Dynamically add the data set VID.SVIDLPA to the LPA with the following MVS command:

   ```
   SETPROG LPA,ADD,DSNAME=VID.SVIDLPA,MASK=VID*
   ```

4. Dynamically add the new subsystem with the following command:

   ```
   SETSSI ADD,SUB=VIDS,I=VIDSSINT
   ```

The CICS VT SSI is now ready to use. Note that any valid name can be chosen for the subsystem, but it is recommended that you use VIDS.

Permanent SSI installation

To install the CICS VT SSI permanently, you must IPL the MVS system. But first you need to make the following changes to SYS1.PARMLIB:

1. Add VID.SVIDLINK to your current PROG*xx* member.

2. Add VID.SVIDLINK to your current linklist either in your PROG*xx* or LNKLST*xx* member.
3. Add VID.SVIDLPA to your current LPALST *xx* member.
4. Add the subsystem to your current IEFSSN*xx* member using the following command:

```
SUBSYS SUBNAME(VIDS) INITRTN(VIDSSINT)
```

The final step to install the SSI is to perform an IPL of the system. If you previously installed the SSI dynamically, you added the VID.SVIDLINK modules to an existing linklisted data set. After the SSI is installed permanently, you should remove these modules from this data set.

Installing CICS VT in CICS

A number of CICS VT objects must be defined in CICS. Jobs are provided to add these objects to the group VID using the batch CSD update facility. In addition, you must define a CICS VT program in the CICS PLT.

If you are using a terminal owning region (TOR) and an application owning region (AOR) or file owning region (FOR), install CICS VT in the CICS region where the application programs that issue the VSAM calls are run.

Your CICS region must be able to connect to DB2 to process SQL calls issued by the CICS VT SQL drivers. If you have not previously enabled the connection, you might have to perform additional DB2 installation activities. In this case, pay particular attention to CICS security.

If you are coding exits in a supported high-level language, you must have CICS definitions for the programs CEEPIPI, IBMPOIOA, IGZCPCO, and IGZEINI.

Installation summary

You must perform the following steps to install CICS VT in every CICS system that will access migrated data sets.

1. Add a CICS VT program to the PLT.
2. Define the CICS VT system objects to CICS.
3. Modify the CICS JCL.
4. Stop and start CICS.
5. Run the CICS VT installation checker transaction VTMC.

These steps are explained in detail next.

Adding a CICS VT program to the PLT

The CICS VT program VIDCINIT enables two CICS exits that are used by CICS VT. This program must be added to the CICS PLT *after* the DFHDELIM entry.

An example of the DFHPLTPI entry needed is in *my* .SVIDCNFG. *custom* member VIDCPLT.

Defining the CICS VT system objects to CICS

The CICS objects that are used by CICS VT are defined in member VIDCDEF in the *my*.SVIDCNFG.*custom* library. This member uses the CSD batch update program DFHCSDUP.

The following CICS objects are defined:

- A PROFILE called VIDPROF and all of the CICS VT transactions and programs.
- TDQUEUE entries for CICS VT tracing.
- A DB2ENTRY for the CICS VT IVP transaction.

All of the CICS VT CICS components are defined in the group VID, which should be added to the CICS startup list.

Modifying the CICS startup JCL

Add DD statements for the library that contain the CICS VT drivers library (created by the install member VIDDDEF), and add VID.SVIDLODE to the DFHRPL concatenation in the CICS startup JCL. The drivers library should be ahead of VID.SVIDLODE and both should be placed as high as possible in the DFHRPL concatenation.

Add the following statements to your CICS JCL:

Table 3. DD statements and functions

DD Statement	Function
`//VIDTRC DD SYSOUT=*`	CICS VT trace output
`//VIDDMP DD SYSOUT=*`	Formatted SQL error report
`//VIDPIPI DD SYSOUT=*`	Messages from HLL user exits
`//VIDTRCD DD SYSOUT=*`	Formatted CICS VT control block report. Also used for the output from the audit status display transaction VTMU.
`//VIDCOMP DD SYSOUT=*`	Output from the dual mode facility

Stop and start CICS

Stop and start CICS to complete the CICS VT installation. You will see the following message issued by the CICS PLT program:

```
VIDCINIT - CICS VT successfully initialised
```

This message is issued every time you start CICS .

Running the installation checker transaction VTMC

When CICS has restarted, run the supplied transaction VTMC. This verifies that all of the required CICS objects have been successfully defined. No input to the transaction is required. After the transaction runs, the screen shown in Figure 5 on page 15 is displayed:

```
--------------------------- CICS VT CICS INSTALLATION CHECK ----------------------


  The following problems (if any) have been found:-

=>  No problems were encountered when checking the CICS definitions required
=>  for CICS VT.
=>
=>
=>
=>
=>
=>
=>




 CLEAR:END
```

Figure 5. VTMC transaction results

If any error is detected, a message is displayed to help you identify the missing component.

CICS DB2 attachment implications

CICS VT loads program DSNCLI from the CICS DFHRPL. This program is supplied in both the CICS SDFHLOAD library and the DB2 SDSNLOAD library, depending on the release of each product you are using. From DB2 Version 6 onwards, DSNCLI is no longer supplied with DB2.

A PROGRAM resource definition must exist for the DB2 module DSNCLI or, alternatively, CICS program auto-install must be active.

Additional configuration actions

There are two further configuration steps to perform:

1. Implement the CICS VT ISPF interface.
2. Make JES JCL procedures available.

The second step is optional.

Implement the CICS VT ISPF interface

The CICS VT ISPF mapping component uses REXX executables in *cvt210*.SVIDEXEC. The REXX member VIDSTART in the library *cvt210*.SVIDEXEC initiates the mapping component.

Start the CICS VT ISPF interface using default parameters with the TSO command:

```
EXEC 'cvt210.SVIDEXEC(VIDSTART)'
```

You can override default data set high-level qualifier and language default values as follows:

```
EXEC 'SYS3.SVIDEXEC(VIDSTART)' 'hlq(cvthlq) lang(K)'
```

CICS VT connects to DB2 , so ensure that the DB2 SDSNLOAD data set is available to your ISPF session.

ISPF and screen readers

CICS VT ISPF screens support screen readers. To avoid pop-up windows, you must ensure that the ISPF environment setting "long message in pop-up" is set to off. Use the ISPF **SETTINGS** command or option 0 from the standard IBM ISPF Primary Option Menu to set the long message option to off.

Make JES JCL procedures available

The JCL procedures VIDASM, VIDDDMG, VIDDIMG, VIDDDLJ, VIDLBR, VIDDB2 and VIDSTGII in *my*.SVIDCNFG.*custom* are used by CICS VT batch utilities. Either copy them to a JES procedure library, or use PROCLIB or JCLLIB statements to reference them.

SYS1.MODGEN considerations

The SYS1.MODGEN macro library data set name is hard-coded in members VIDGDMEX and VIDASM in the *my*.SVIDCNFG.*custom* library. If this data set has the name SYS1.AMODGEN in your installation, manually update members VIDGDMEX and VIDASM.

Additional DB2 configuration actions

Certain CICS VT operations depend on characteristics of DB2 at your site.

1. The CICS VT automated mapping facility uses the DB2 sample program DSNTEP2 to execute the generated DDL.
2. The generated JCL to create the CICS VT driver modules includes DB2 bind statements.

Tailoring that is specifically related to aspects of your DB2 subsystem is required.

Define the sample DB2 program DSNTEP2

The automated mapping feature generates DDL, and then executes it using the DB2 sample program DSNTEP2. CICS VT assumes that the name of the DB2 plan for this program is DSNTEP2. If this is not the plan name you use, manually update the member VIDDDLTS in *my*.SVIDCNFG.*custom* library. Also verify that the correct data set is specified in the LIB statement, and update if required.

Customizing skeleton bind parameters

CICS VT uses a skeleton member to generate the bind parameters for DB2 packages and plans. The default member is called VIDBIND and is in *my*.SVIDSAMP.*custom*. Comments explaining the tailoring options are included in this member.

See “DB2 considerations” on page 58 for further discussion concerning the VIDBIND member.

If you are going to generate read-only DDMs, you will have to create an additional skeleton member in *my* .SVIDSAMP. *custom* . Specify this new member name in the second mapping defaults panel.

Optionally implement a new DDM driver naming exit

The name of the DDM module is based on the VSAM file name and controlled by the CICS VT exit VIDDDMEX. If you choose a DIM name of 7 characters or less, the default VIDDDMEX adds the character '#' to the end. For example, if you select MYFILE as the name of a DIM, the DDM name using the default VIDDDMEX becomes MYFILE#.

Avoid using 8-character DIM names. For example, if you select DIM names MYFILE01 and MYFILE02, the DDM in both cases becomes MYFILE0# and the second DDM you generate will overwrite the first. If you have to use 8-character DIM names, you must develop your own version of VIDDDMEX to avoid this.

The default VIDDDMEX module is supplied in load form in *cvt210*.SVIDLODE. The exit source and JCL to assemble and link the default module is supplied in *my*.SVIDCNFG.*custom* member VIDGDMEX.

Steps in the IVP will fail if you change the default exit. If you decide to change the default exit, wait until all parts of the IVP have been run successfully.

Performing the installation verification procedure (IVP)

A sample VSAM file called VIDKSDS is supplied with CICS VT. This file has been mapped to CICS VT and the DIM and DDM driver modules are shipped in VID.SVIDLODE. To verify that the CICS VT installation is complete, two batch programs and one CICS program process the file. Some manual tailoring is required to run the batch IVP programs.

The final verification step tests that the ISPF component is correctly enabled and that you can successfully generate a DIM and DDM.

Batch IVP

Ensure that you have completed the installation step “Installing the batch subsystem” on page 12. Edit the *my*.SVIDCNFG.*custom* member VIDIVPJ1 and change the following parameters to the correct values for your system:

- Change @KSDSNAME to a valid data set name (which is created when you run the VIDIVPJ1 job).
- Change @VOLUME to the disk where the VSAM data set will be created.

Submit the job for processing. The data set that is created by VIDIVPJ1 is only used for the purposes of the batch IVP. It can be deleted when the IVP is completed. The JCL assumes that the DB2 SDSNLOAD library is in the linklist. If this is not true, add the library to your STEPLIB concatenation.

After VIDIVPJ1 runs successfully, tailor and run the two batch IVP programs in *my*.SVIDCNFG.*custom*. Member VIDIVPCO runs a COBOL program and VIDIVPL1 runs a PL/I program. If you do not use PL/I, you do not need to run this program.

Manually change the following parameters:

- @KSDSNAME to the data set name you used in member VIDIVPJ1.
- @SSID to your CICS VT subsystem name.

The IVP jobs should end with a condition code of zero. You can also check that the jobs were successful by examining the output report written to the IVPREPT DD statement. There should be 20 lines of output.

CA Top Secret users

The CICS VT COBOL batch IVP program VIDIVPCO may terminate abnormally with an S0C4 error in sites that use CA Top Secret. Review the IVPREPT DD statement, and if there is output then it is likely that you are a victim of this issue.

Apply the Top Secret fix BIT8325 to resolve the S0C4 error.

ISPF IVP

The *my*.SVIDSAMP.*custom* members VIDIVPC and VIDIVPP are copybooks, and include instructions to automatically map a version of the VIDKSDS IVP data set. VIDIVPC is COBOL and VIDIVPP is PL/I.

Select either of these members and follow the embedded instructions.

When the ISPF IVP has successfully completed, the automated mapping facility has been correctly customized and is available for you to use for your own VSAM files.

CICS IVP

The *my*.SVIDCNFG.*custom* member VIDCDEF adds the CICS IVP program and transaction to the CICS CSD. You must install the group VID before you can run the CICS transaction.

The CICS transaction is VTMV. When you run this transaction, the following screen is displayed:

```
---------------------------  CICS VT IVP TRANSACTION   ---------------------------



Enter ITEM number and press ENTER                245021

(This transaction will verify that CICS VT is installed and functioning
correctly in this CICS system)


Please Enter the ITEM number


 PF3:END  CLEAR:REFRESH SCREEN
```

Figure 6. VTMV IVP transaction screen

Enter `245021` as the value for the ITEM number and press Enter. The program issues a call to the CICS VT sample VSAM data set VIDKSDS, which has been migrated to DB2. The VSAM calls are processed by CICS VT in DB2, and details for this item number are displayed on the screen.

Depending on the security that you have enabled in CICS, you might have to sign on to CICS before running VTMV to establish a thread to DB2. If you don't sign on to CICS, the DB2 connection might be unsuccessful and result in a CICS transaction abend. Typically, this would be an ASP7 abend.

CICS VT drivers IVP

The CICS VT mapping data for the sample file VIDKSDS that is processed by the IVP programs is shipped in the *my*.SVIDCNFG.*custom* member VIDMAPDA. To be able to perform the IVP for the CICS VT drivers, run this member using DB2 SPUFI.

When the mapping data is inserted, complete the IVP process by generating the DIM and DDM drivers for the file VIDKSDS. There are three steps:

1. Start the CICS VT ISPF application. This is explained in "Implement the CICS VT ISPF interface" on page 15.
2. Define parameter default values. This is only required the first time you use the ISPF application. This task is covered in "Defining mapping defaults" on page 32.
3. Generate the CICS VT driver modules by following the instructions in "Generating CICS VT runtime modules" on page 53.

You should generate the DIM and DDM drivers to the driver data sets that you created earlier in "Creating CICS VT user data sets" on page 12. Then rerun the batch IVP jobs, but ensure that the newly generated drivers library is added as the first STEPLIB data set.

Additional CICS transactions

There are four other supplied CICS transactions that you can use. Their purpose and the circumstances in which you use them are described next.

VTMD - displaying the CICS VT DST

The CICS VT data set table (DST) contains a list of all the VSAM files that are defined to CICS VT. Use the VTMD transaction to display a list of these files, shown in the following screen:

```
        08/23/11     ==>  CICS VT File Definitions  <==     12:52:52

Filename  DIM name   Org   Status            MStatus Operations    Compare
ACCTFIL   ACCTFIL          ENA INI           ACT     R U A B D     REP   N
ACCTFILE  ACCTFIL          ENA INI           ACT     R U A B D     REP   N
ACCTNAM   ACCTX01          ENA INI           ACT     R U A B D     REP   N
ACCTNAME  ACCTX01          ENA INI           ACT     R U A B D     REP   N
APPLCTL   APPLCTL          ENA INI           ACT     R U A B D     REP   N
CAP1BR    CAP1BR           ENA INI           ACT     R U A B D     REP   N
CLE0      CLE0             ENA INI           ACT     R U A B D     REP   N
CLE00     CLE00            ENA INI           ACT     R U A B D     REP   N
CLE001    CLE001           ENA INI           ACT     R U A B D     REP   N
CLE001A   CLE0X01          ENA INI           ACT     R U A B D     REP   N
CLE001AA  CLE001AA         ENA INI           ACT     R U A B D     REP   N
CLE001B   CLE0X02          ENA INI           ACT     R U A B D     REP   N
CLE001C   CLE001C          ENA INI           ACT     R U A B D     REP   N
CLE001F   CLE001F          ENA INI           ACT     R U A B D     REP   N
CLE001FA  CLE001FA         ENA INI           ACT     R U A B D     REP   N
CLE001FF  CLE001FF         ENA INI           ACT     R U A B D     REP   N
CLE002    CLE002           ENA INI           ACT     R U A B D     REP   N
CLE002B   CLE002B          ENA INI           ACT     R U A B D     REP   N
More
PF3=end          PF8=fwd                          SYSID=S650 APPLID=CICSTS32
```

Figure 7. VTMD display CICS VT DST

No user parameters are required for the VTMD transaction. Optionally, you can specify a full or partial file name to position the display. For example:

```
VTMD ITE
```

This will position the file ITEMFL as first file displayed.

VTMI - subsystem control transaction

When you migrate a VSAM file, you must add it to the CICS VT DST so that CICS transactions can access the data in DB2. To activate a change to the table, the CICS VT CICS subsystem must stop and restart. Use the VTMI transaction for this purpose.

Specify one of the following parameters for the VTMI transaction:

STAT This shows the status of the CICS VT interface.

START
This starts the CICS VT interface.

STOP This stops the CICS VT interface.

RESTART WARM
This stops and restarts the interface and loads the DST. The migration status of each file changed by the VTMM transaction persists.

RESTART COLD
This stops and restarts the interface and loads the DST. The migration status of all files is reset to the value of MSTATUS in the DST.

To add a new file to the DST, follow these steps:

1. Add the new file entry to VIDCDTAB.
2. Assemble the table into the DFHRPL library.
3. Run the transaction VTMI RESTART to stop and restart the interface. This also issues a NEWCOPY for the new VIDCDTAB module.

If the MSTATUS of the new file is ACT, all calls to this file from CICS programs now access the data in DB2. If the MSTATUS is INACT, calls continue to be processed in VSAM. If you run the VTMI transaction with the STOP keyword, all VSAM calls for all files are processed by CICS.

The DST is explained in full in "Adding a new data set to the CICS VT data set table" on page 104.

VTMM - switching the file migration status

Use the VTMM transaction to switch the CICS VT migration status of a specific file. The format of the transaction is:

```
VTMM file-name ACT or
VTMM file-name INACT
```

The migration status of the file established by the VTMM transaction persists until any of the following events:

- It is reset by VTMM *file-name* ACT or INACT
- It is reset by VTMI RESTART COLD
- CICS is stopped and restarted

You must specify a single *file-name* with the VTMM transaction.

The DST is explained in full in "Adding a new data set to the CICS VT data set table" on page 104 .

VTMA - dual mode control transaction

The dual mode facility (DMF) causes calls to be processed against a VSAM data set and the migrated data in DB2. The VTMA transaction controls the DMF. You must specify one of the following parameters with VTMA:

STAT This shows the status of DMF.

START
This starts DMF.

STOP This stops DMF.

RESTART
This restarts DMF and is only required when applying software maintenance.

The DMF is explained in full in Chapter 6, "The dual mode facility (DMF)," on page 111.

VTMU - audit status report transaction

The VTMU transaction generates the CICS VT audit status report. The report is written to the VIDTRCD DD statement.

IBM may ask for this transaction to be run.

Upgrading from CICS VT version 1.2

If you are upgrading from an existing version 1.2 system, there are a number of actions to be performed, which are described in this section.

The version 1.2 APAR PM16132 must be applied before you upgrade to version 2.1. The CICS VT control tables and mapping data must also have been migrated according to the instructions with PM16132. You can confirm this by running a DB2 query against the catalog table SYSIBM.SYSCOLUMNS for the CICS VT control table *cvt*.DSN_MAP and verifying that the COLTYPE=VARCHAR and LENGTH=128 for the column VIDSPCRE.

You must run the installation customization using the version 2.1 data sets. This ensures that version 2.1 objects are tailored for your environment

Compatibility between versions 1.2 and 2.1

Existing CICS VT application objects, such as your DIM and DDM drivers modules, are compatible between version 1.2 and version 2.1. Drivers generated with version 1.2 will run in a version 2.1 system. Drivers generated with version 2.1 will run in a version 1.2 system. Version 2.1 uses the version 1.2 mapping tables.

There are no required changes to your existing batch JCL. However, if the DD statement for your migrated VSAM files includes a DSN= parameter, new version 2.1 function is invoked automatically. This is explained in "Upgrading the batch subsystem" on page 23.

Version 1.2 FBE and IRD exits are compatible and no changes or recompiles are required for version 2.1.

There is no advantage in maintaining separate version 1.2 and version 2.1 systems. The recommendation is to use version 2.1 as soon as you have completed the appropriate upgrade tasks.

Upgrading system components

A subset of the installation steps are required to upgrade an existing version 1.2 system to version 2.1.

You must complete three actions to upgrade CICS VT system components:

1. *my*.SVIDCNFG.*custom* member VIDB2BND binds the version 2.1 DB2 packages and plans. Refer to "Bind CICS VT packages and plans" on page 11 for instructions. The package VIDMIGG2 is new in version 2.1.

   All CICS VT system DBRMs have a version ID related to the product version. If you have to fallback from version 2.1 to version 1.2, you do not have to run version 1.2 *my*.SVIDCNFG.*custom* member VIDB2BND. When you have fully migrated to version 2.1, the version 1.2 packages can be freed. The version ID for version 1.2 is CVT120.
2. The existing version 1.2 CICS VT objects defined in CICS are used by version 2.1. New CICS VT objects associated with the VTMU transaction must be to be defined to CICS. Member VIDCDEF2 in *my*.SVIDCNFG.*custom* contains the input to the batch CSD update utility to define the new objects. Add them to the same CICS group as the existing version 1.2 objects. The default group name is VID.
3. There are 5 new JCL procedures in version 2.1. The version 1.2 procedure VIDPRE is replaced by procedure VIDDB2 in version 2.1. Refer to "Make JES JCL procedures available" on page 16 for a discussion on implementing JCL procedures.

Upgrading ISPF components

All five version 1.2 ISPF data sets are replaced by equivalent version 2.1 data sets.

There are two key differences in the version 2.1 ISPF environment.

- The SVIDPLI *s* library members are compiled DTL. The look and feel of the version 2.1 panels is similar but not identical to version 1.2.
- The main menu has an additional option for the automated data migration feature.

Several SVIDLLIB programs access DB2. Ensure *my*.SVIDCNFG.*custom* member VIDB2BND has run successfully before you use the version 2.1 ISPF component.

If you modified the default skeleton bind parameter member for version 1.2, you may have to replace the default version 2.1 member. Refer to "Additional DB2 configuration actions" on page 16 .

Note that you cannot have version 1.2 and version 2.1 active in the same ISPF session at the same time.

Upgrading the batch subsystem

Actions are required to upgrade an existing CICS VT version 1.2 system to version 2.1. Prior to performing the required actions, it is essential to understand the difference between version 1.2 and version 2.1 subsystems.

Prior to upgrading

Certain JCL changes have always been required to enable a batch job to access a VSAM data set migrated to DB2 with CICS VT. One of the required changes applies to the DD statements that define the migrated VSAM data sets; you must add a SUBSYS statement and appropriate CICS VT parameters. When the SUBSYS parameter is added, the CICS VT subsystem handles all VSAM calls to this DD.

The version 1.2 subsystem ignores other JCL parameters in the VSAM data set DD statement. The version 2.1 subsystem has a new migration status facility. This facility is sensitive to the existence of the DSN parameter. For your original version 1.2 migration, you may have added SUBSYS and without removing other DD statement parameters, including DSN=*original.VSAM.dataset*. If the version 2.1 subsystem detects a DSN parameter, the migration status facility is invoked. A new version 2.1 module VIDFMSP is loaded. If you are not using the version 2.1 library in your JOBLIB/STEPLIB, your application program abends and the message VIDSS204 referring to module VIDFMSP is issued.

You should review the JCL for all your files migrated with version 1.2 and remove all DSN= parameters prior to upgrading to version 2.1.

Prior to upgrading, you should let any existing batch jobs using CICS VT complete, and prevent any new CICS VT batch jobs from starting.

Upgrading subsystem initialization modules

The batch subsystem uses the libraries *cvt*.SVIDLPA and *cvt*.SVIDLINK. The same SVIDLPA and SVIDLINK module names are used for initializing version 1.2 and version 2.1 subsystems. These modules are upward and downward compatible. For each instance of a CICS VT subsystem the following method is recommended:

- The version 1.2 SVIDLPA and SVIDLINK libraries or library names are reused for the version 2.1 libraries.
- The version 1.2 batch subsystem name is used for the version 2.1 batch subsystem.

Use IEBCOPY to copy the version 2.1 SVIDLPA and SVIDLINK libraries over the version 1.2 libraries.

If you want to keep separate version 1.2 and 2.1 subsystems, you can define a new subsystem by following the instructions in “Installing the batch subsystem” on page 12.

Updating subsystem modules

SVIDLPA and SVIDLINK modules are only used during subsystem initialization. CICS VT subsystems defined in your system PARMLIB library are initialized at a system IPL. New CICS VT subsystems defined dynamically are initialized immediately. The CICS VT utility VIDUTILU dynamically updates an existing subsystem by reloading the LPA modules. This means that you can enable a version 2.1 subsystem and immediately exploit the batch migration status facility.

To update an existing version 1.2 subsystem to version 2.1, the first step is to back up your existing version 1.2 SVIDLPA and SVIDLINK libraries. Next, copy the entire version 2.1 SVIDLINK and SVIDLPA libraries to the existing version 1.2 libraries using IEBCOPY or an equivalent utility.

Activate the version 2.1 libraries with the following z/OS commands:

```
F LLA,REFRESH
SETPROG LPA,ADD,MODNAME=VIDSSLPA,DSNAME=cvt.SVIDLPA
SETPROG LPA,ADD,MODNAME=VIDSSLPB,DSNAME=cvt.SVIDLPA
```

The final step is to run the *my*.SVIDCNFG.*custom* member VIDUTILU. Note that the VIDUTILU parm statement assumes the default CICS VT subsystem name of *VIDS*. Change this if required.

These dynamic changes are hardened at the next system IPL.

If you have decided to configure a new subsystem for version 2.1, follow the instructions in "Installing the batch subsystem" on page 12. The VIDUTILU utility is not required for a new subsystem.

Configuring CICS VT on additional systems

It is only necessary to perform the complete configuration process on one system, normally a test or development system. All of the mapping and initial testing should take place here. To run CICS VT on other systems such as a quality assurance or production system, a subset of configuration tasks are required.

These are explained in:

- "Installing the batch subsystem" on page 12
- "Installing CICS VT in CICS" on page 13

You should also copy the following application components to the additional system:

- The DDL for the DB2 objects. You might have to change the DB2 table creator, storage group and database names to suit your environment.
- The CICS VT drivers and DBRMs. The DBRMs must be bound into packages, and plans must be bound for batch access.
- CICS VT exits and DBRMs if appropriate. Bind the DBRMs into the same collection as the DDM driver packages.
- The data migration jobs VIDLOAD and VIDUNLOAD, as well as the DB2 load utility JCL and control cards. You might have to update the DB2 SSID in some of these jobs.
- Other DB2 utility jobs, such as IMAGE COPY and RUNSTATS, as required.

You might also want to copy the JCL procedures that include the modifications to enable CICS VT.

Chapter 3. Defining VSAM files to CICS VT

This section explains how you use CICS VT to migrate VSAM files to DB2 and is aimed at DBAs and application developers. You should read this information thoroughly before you start to migrate your first file.

Your first file should be of low importance and ideally standalone. The more files you migrate, the more familiar you become with the capabilities of CICS VT and the data migration process. You might decide to go back and change some aspects of a file you have previously migrated, to change the DB2 table design for example. CICS VT allows you to do this. By starting with a simple, low-risk file, going back to make these changes will not incur a significant retesting overhead.

There are two different methods to migrate a VSAM data set using CICS VT:

- Perform the migration manually
- Use the automated mapping facility

You should be able to migrate most files using the automated mapping facility, but this depends on how the file is defined in the copybook. This chapter explains how CICS VT helps you decide which method to use, and what needs to be done in both cases.

Functional overview

CICS VT works by intercepting runtime calls to VSAM files from your application programs.

For batch programs, this interception is performed using an MVS subsystem. In CICS, the interception is handled by a global user exit (GLUE). The call interception is referred to as the " CICS VT runtime component".

At run time, CICS VT uses two driver modules for each migrated VSAM data set:

Table 4. Driver modules

Driver module	Description
Data set Information Module (DIM)	This defines the relationship between the VSAM record structure and the DB2 table it relates to.
Data set Driver Module (DDM)	This contains the necessary SQL to access the DB2 table.

These modules are generated by standard CICS VT ISPF dialog functions. These functions are part of the " CICS VT mapping component".

There is a third component of CICS VT called the " CICS VT data migration component". This component deals with all aspects of the initial conversion of your VSAM files to DB2 tables.

The overall architecture of CICS VT can be viewed as these three separate components, which are discussed in more detail throughout the remainder of this section.

The runtime component

Although there are differences between the batch and CICS runtime components in the way that they intercept application program VSAM calls, functionally they are equivalent. This component performs a number of functions:

- It intercepts the VSAM calls within your application programs.
- In CICS programs, it analyzes the call to decide if the target file has been migrated to DB2. If the file has been migrated, it loads and processes the appropriate CICS VT driver modules. In batch programs, CICS VT is enabled via JCL changes.
- It accesses DB2 to process the data.
- It builds the equivalent of the VSAM or CICS return codes that drive your program logic.

The mapping component

CICS VT has to understand the structure of the records in a VSAM file and how this relates to the structure of the data in DB2. The relationship between these structures is achieved through a CICS VT process called mapping.

Although there are two separate mapping methods, an automated method and a manual method, both achieve the same purpose. You should understand the mapping component regardless of the method you use.

Figure 8 shows the IDCAMS DEFINE CLUSTER statements for a KSDS data set and is used here to illustrate the mapping process.

```
DEFINE    CLUSTER                                -
            NAME(VID.VIDKSDS)                    -
            INDEXED SHR(2 3) FREESPACE(30 10))   -
          DATA                                   -
            (NAME(VID.VIDKSDS.DATA)              -
            KEYS(6 0) VOL(CTC002) RECSZ(80 80)   -
            CISZ(16384) CYL(1 10))               -
          INDEX                                  -
            (NAME(VID.VIDKSDS.INDEX)             -
            CISZ(2048) VOL(CTC002) CYL(1 1))
```

Figure 8. Sample define cluster for a KSDS

Next, look at the COBOL copybook for this data set.

```
01 ITEM-DEF
   02 ITEM-NUMBER                PIC X(6).
   02 ITEM-NAME                  PIC X(12).
   02 ITEM-DESC.
      03    ITEM-COLOUR           PIC X(6).
      03    ITEM-WEIGHT           PIC X(4).
      03    ITEM-COST             PIC S99999V99 COMP-3.
   02 ITEM-REORDER-QUANTITY    PIC S999.
   02 ITEM-SUPPLIER-CODE     PIC X(3).
   02 ITEM-DATE-FIRST-SHIP    PIC S9(9) COMP-3.
   02 ITEM-SHELF-LIFE      PIC X(2).
   02 ITEM-DESCRIPTION     PIC X(35).
```

Figure 9. Sample copybook

The design of the DB2 table for this VSAM file depends on its planned usage outside of CICS VT. For example, you might want to migrate a data set to DB2 to share the data between your online programs and your batch housekeeping

utilities. The simplest DB2 design to achieve this objective is as follows:

```
 CREATE TABLE VID_ITEM
       (ITEM_KEY                  CHAR(6)
       ,ITEM_DATA                 CHAR(74)
   (PRIMARY KEY(ITEM_ROOT,ITEM_NUMBER))
IN ITEMDB.ITEMTS;
```

Figure 10. DB2 DDL for a key and data solution

In this design, which is referred to as a key and data design, no copybook is used. An advantage of this design is that potential data inconsistencies are eliminated because DB2 imposes no rules on character data. This table design is appropriate if you simply want to convert a VSAM data set to DB2 so that the data is under the control of a DBMS. Normally, there are no additional requirements to access tables using this design.

The obvious disadvantage of this design is that the value of the table from an SQL perspective is minimal. You could not use this design if you plan to provide end user access to the migrated tables, although CICS VT allows you to change the design of the DB2 table after the initial migration.

If you plan to write new programs to access the data after it has been migrated to DB2 or to provide ad hoc access to end users, a copybook based table design is required. For the sample copybook shown in Figure 9 on page 26 the DB2 design could look as follows:

```
 CREATE TABLE VID_ITEM
       (ITEM_NUMBER               CHAR(6)
       ,ITEM_NAME                 CHAR(12)
       ,ITEM_COLOUR               CHAR(6)
       ,ITEM_WEIGHT               CHAR(4)
       ,ITEM_COST                 DEC(7,2)
       ,ITEM_REORDER_NO           SMALLINT
       ,ITEM_SUPP_CODE            CHAR(3)
       ,ITEM_DATE_FSHIP           DATE
       ,ITEM_SHELF_LIFE           CHAR(2)
       ,ITEM_DESCRIPTION          CHAR(35)
   (PRIMARY KEY(ITEM_NUMBER))
    IN ITEMDB.ITEMTS;
```

Figure 11. DB2 DDL solution using a copybook

There are a number of points to note about this DB2 table design:

1. The DB2 primary key is always the key of the KSDS data set. For an RRDS data set, the 4-byte relative record number becomes the DB2 primary key.
2. The DB2 primary key must be the first column or columns in the table.
3. Each copybook field shown in Figure 9 on page 26 corresponds to a DB2 column. The copybook group field ITEM-DESC can optionally be added as an SQL COMMENT on the DB2 column for your documentation purposes.
4. The copybook field ITEM-REORDER-QUANTITY is signed, zoned decimal with an attribute of PIC S999. The DB2 column it maps to is data type SMALLINT. Data conversion is automatically handled by CICS VT. By using the data re-engineering capability of CICS VT in this way, SQL arithmetic functions can be performed against this column. This could be useful for end user queries.
5. The copybook field ITEM-DATE-FIRST-SHIP is packed decimal. The DB2 column it maps to is data type DATE. CICS VT handles data conversion automatically, but the layout of the field must be defined to CICS VT. For example, this could be CCYYMMDD or MMDDCCYY. CICS VT converts the

field value from its packed decimal form to the correct DB2 DATE value when a record is inserted or updated. For retrieval calls, CICS VT converts the DATE data from DB2 into a packed decimal value according to the date field layout.

Using the re-engineering capabilities of CICS VT enhances the business value of your data once it is migrated to DB2. This can be especially useful for end users. However, it assumes that the VSAM data field values are consistent with the copybook attributes. Inconsistent data will increase the time to perform data migration.

You should consider the requirements for each VSAM file, before deciding on the most appropriate DB2 table design.

The data migration component

CICS VT is a suite of tools that cover the entire VSAM to DB2 migration process, including the key component of data migration. To migrate data from a VSAM file to a DB2 table, you must run a series of CICS VT batch utility jobs.

Some of these batch jobs use utilities provided with CICS VT and some use standard DB2 utilities such as LOAD, RUNSTATS, and REORG. In addition, you can use the DSNTIAUL DB2 sample program to generate the DB2 LOAD control cards.

When you are migrating the data, CICS VT automatically performs data re-engineering, according to how you have mapped the file. For example, if you have a field containing date information and you map this field to a DB2 DATE column, the data migration process builds the appropriate DB2 column values so that the data presented to the DB2 LOAD utility is correct.

An automated data migration facility generates the JCL and control cards for the each migration step.

Typical CICS VT migration process

The migration methodology is the same for each VSAM file you migrate, regardless of aspects such as the attributes of a file or how it is used. The key steps in the migration process are as follows:

1. Identify which file or files are to be migrated.
2. Analyze the files to determine DB2 table designs.
3. Create the DB2 tablespace, table and index objects.
4. Map the file or files.
5. Migrate the data from VSAM to DB2.
6. Test the mapping.
7. Test application programs.
8. Cutover to production.

In many cases, steps 2-5 are performed automatically, but here they are explained in detail to help you understand the complete migration process.

Step 1 - Identify the files to be migrated

Migration is an iterative process. This first step is to decide which file or files you want to migrate at the same time. You must choose a unique name for each base cluster data set. This is referred to in CICS VT as the DIM. You should use a

maximum of 7 characters for the DIM name. The reason for this is explained in "The DDM driver module name" on page 59. You can migrate files individually, or collectively. CICS VT refers to this as a migration unit. CICS VT does not impose any rules; make your decision based on factors such as the criticality of the file, its size, and the required availability.

You might decide to migrate the file at the same time as you make other routine application changes, in order to reduce application testing.

Step 2 - Analyze the files

You should analyze the file or files you are going to migrate. There are a number of reasons for doing this:

- Each alternate index path must be mapped. Analysis of the file establishes the characteristics of each alternate index path.
- Identify all of the application programs that access the file. This allows you to develop a test plan. It also allows you to identify all of the batch JCL procedures that need to be modified to enable CICS VT access to a migrated data set.
- Some of the fields or records in the file might be obsolete. You should choose whether you want to migrate potentially obsolete data to DB2 or take the opportunity to clean up the file.
- Identify future requirements for the file, after it has been migrated to DB2. For example, you might want to add new columns to the table to support enhancements that you plan to make to the application.
- Some files might have multiple copybooks. You need to decide how to handle this.
- CICS VT supports multiple START BROWSE processes in CICS, up to a limit specified as a CICS VT user option. This is covered in "Generating CICS VT runtime modules" on page 53. You should establish the maximum number of concurrent browse processes for each file to be migrated.

When you have analyzed your file or files, you will know the basic design of the DB2 tables that will be used and which batch and CICS programs will need to be tested.

Step 3 - Design and create the DB2 objects

The SQL data definition language (DDL) for the DB2 tables and indexes must be produced in this step. This can be done manually or by using the automated mapping facility. In most cases, your DB2 table will look similar to your file copybook, based on the outcome of the file analysis step.

However, there might be characteristics of the file that influence your ultimate table layout. For example, it is best practice to group together those columns that are most frequently updated to optimize DB2 performance and logging.

The relative position of a field in your copybook does not dictate the position of the column within the DB2 table. You can arrange the columns in your DB2 table to optimize DB2 . Every CICS VT DB2 table must have a primary key that must be the first column or columns in the table, and a primary index must be created. This is normally designated as the clustering index.

Step 4 - Map the data set to CICS VT

CICS VT provides an automated mapping facility and a manual mapping facility. The automated mapping facility reads the base cluster information, and then

analyzes the copybook to determine if the data set can be automatically migrated or not. Data sets with complex copybooks, such as nested redefined structures, must be manually mapped and might require CICS VT user exit APIs to achieve the target design in DB2 . When the mapping is complete, you generate the CICS VT runtime driver modules, the DIM and DDM.

Step 5 - Migrate data

Data migration uses a combination of CICS VT and DB2 utilities. It is a 3-step process as follows:

1. Unload the VSAM file to a sequential data set by using the CICS VT VIDUNLOD program.
2. Convert the sequential data set into a DB2 LOAD format data set using the CICS VT VIDLOAD program. CICS VT services are invoked at this stage to handle any field re-engineering, according to how the data set is mapped.
3. Use the DB2 LOAD utility to load the output data set from step 2 into the DB2 table.

You should use the automated data migration facility to generate the CICS VT batch job utility and control cards.

The data migration process can involve some degree of data cleansing, particularly where you are mapping fields to DECIMAL, INTEGER, SMALLINT, or DATE column types.

Step 6 - Test mapping

Before you start to test your application programs, verify that the mapping is correct. This is especially important if you are performing any data re-engineering. To do this, run the VIDUNLOD program against the migrated data set and compare this with the file created in the first activity of step 5.

Step 7 - Test application programs

This is likely to be the longest phase in your migration cycle. Normally, it is not necessary to test every program that accesses the file you have migrated, but this depends on the criticality of the file.

The amount of testing you do should be a factor of the amount of data re-engineering you are performing. You should plan to spend more time testing files that include significant re-engineering.

Step 8 - Cutover to production

This step performs the conversion of your production files. During the cutover process, the VSAM file or files are not available to application programs until you have successfully completed testing. The time for this could range from a few minutes to many hours.

To minimize the time when the data is unavailable, preparation as possible in advance of the cutover. This preparation includes tasks such as creating the DB2 objects and binding the DB2 packages. See Chapter 7, “Production cutover,” on page 119 for a detailed list of cutover tasks.

Automated mapping versus manual mapping

Mapping provides CICS VT with the information it requires to generate the driver modules that process the VSAM calls issued by your application programs. There are three parts to mapping.

The first part is to create the DB2 tables that will contain the VSAM data set data and the indexes that support access to the data. SQL is needed to create these objects. The second part is to establish the relationship between the copybook fields and the DB2 columns they relate to. The third part is to generate the runtime modules for each data set.

The automated mapping facility analyzes the application copybook and automatically generates the DB2 DDL, the mapping data, and the runtime modules. The automated mapping facility also automatically maps alternate indexes, generates the DB2 DDL for the appropriate secondary indexes, and generates the CICS VT drivers. In the manual mapping method, you need to create the DDL, manually map the data set, and generate the runtime modules. You must identify each alternate index and manually map it using a CICS VT batch utility.

You cannot use the automated method if there is no copybook for the data set.

The automated mapping facility identifies copybook fields that may require a user action, such as truncated column names and redefined fields. You must action each identified field before you can continue the mapping process. The automated mapping facility does not support the mapping of one VSAM data set to more than one DB2 table.

If you map a file using the automated facility and decide to change the mapping at a later date, use the manual mapping facilities to make the changes. Making mapping changes to files that were mapped automatically may adversely affect alternate index mapping. You may have to make the same manual mapping changes to alternate index mappings for indexes that were initially mapped automatically.

For more information about automated mapping read “The automated mapping facility” on page 59 . You should read this before you use the automated mapping facility to migrate a data set.

You must enable the ISPF mapping component before you can use either the manual or the automated mapping facility. This is covered in “Implement the CICS VT ISPF interface” on page 15 .

Mapping a VSAM file to CICS VT manually

This section explains how to manually map a VSAM file to CICS VT.

You can map many files using the automated mapper, but in certain circumstances it may be necessary to modify generated mapping. For example, if you encounter inconsistent numeric data during the data migration phase, one solution is to change the field type to character and make the DB2 column type CHAR. Another solution is to correct the inconsistent data in VSAM. You must use the manual mapping dialogs if you need to make changes to the generated mapping.

Because of the potential need to change mapping, it is important to understand the manual mapping process. (For details of the automated mapping method, see “The automated mapping facility” on page 59).

There are some slight differences in the way that a KSDS data set is mapped compared to an RRDS data set. The screens and explanation that follow apply to a KSDS data set. The differences when mapping an RRDS data set are detailed in “Mapping an RRDS data set” on page 47 . The automated mapping facility handles the differences for you.

Alternate index paths must be mapped, although the mapping process is different from mapping base cluster data sets. Alternate index mapping is covered in “Mapping alternate indexes” on page 51 .

Defining mapping defaults

You map data sets using the CICS VT ISPF dialogs. The ISPF dialogs are invoked by the VID.SVIDEXEC member VIDSTART. The first time you start the mapping application, you need to define a number of default mapping values. You do this by selecting option D from the CICS VT main menu screen.

```
 ---------------------------- CICS VT: Main menu ----------------------------


Select Option ===> D_____________________________________________________

      1 - Auto mapping facility

      2 - Manual mapping facility

      3 - Generate migration jobs for DIM : ________ (Full name only)

      D - Defaults









Press: Enter to continue   PF3 to exit   PF1 for Help
```

Figure 12. CICS VT main menu

Specifying mandatory defaults panel 1

There are four defaults panels. The options on panels 1 and 2 are common to the automated and the manual mapping methods. All fields are mandatory. The fields for the first defaults panel are shown in Figure 13 on page 33.

```
 ---------------------- CICS VT: Defaults panel 1 of 4 ----------------------
Command ===> ____________________________________________________________________

 Enter defaults for the generation of the CICS VT drivers:-

 DB2 subsystem               ==> db2t_____________________________________
 SDSNLOAD library (optional) ==> _____________________________________

 CICS VT load library        ==> cvt210.SVIDLOAD_____________________________

 Output Libraries for CICS VT drivers
   load library              ==> appl.drivers.load_____________________
   DBRMLIB                   ==> appl.drivers.dbrm_____________________

 BIND model in SVIDSAMP      ==> vidbind_____________________________________

 DB2 owner for driver bind   ==> dbadm_____________________________________

 No of concurrent threads    ==> 01        Value 01-99

 Output device type          ==> SYSDA___  Generic device type (example:SYSDA)

 Enter=Confirm   PF1=Help   PF3=Exit
```

Figure 13. CICS VT defaults panel 1 of 4

The field descriptions are as follows:

Table 5. Field descriptions

Field	Description
DB2 subsystem	The name of the DB2 subsystem where the data sets are mapped.
SDSNLOAD library	The name of the DB2 SDSNLOAD library. This field is optional if SDSNLOAD is defined in the z/OS linklist.
CICS VT load library	The name of the load library containing the CICS VT system modules.
Output load library for CICS VT drivers	The name of the load library that you created during configuration and setup for the CICS VT driver modules for each migrated data set. See Figure 3 on page 9 .
DBRMLIB for CICS VT drivers	The name of the DBRM library that you created during configuration and setup for the CICS VT driver module DBRMs for each migrated data set. See Figure 3 on page 9 .
BIND model in SVIDSAMP	The name of the skeleton bind member in your *my* .SVIDSAMP. *custom* data set. This member contains default DB2 values that are used during the binding of the CICS VT read-write DDM packages and plans. See “DB2 considerations” on page 58 .
DB2 owner for driver bind	The value of the **OWNER** parameter for the DB2 package and plan bind.
No of concurrent threads	The maximum number of concurrent threads per user for this data set. See “Number of concurrent threads” on page 34 .
Output device type	The name of a generic disk device type, which is used by various CICS VT utilities.

The default values you specify apply to all of the drivers that you generate. The **DB2 owner for driver bind** and **No of concurrent threads** options can be changed at an individual DIM level, as shown in Figure 35 on page 55 .

Note that if you subsequently change the value of the DB2 subsystem, you must exit and restart the ISPF application before the new subsystem value takes effect.

Number of concurrent threads

CICS VT supports multiple access control blocks (ACBs) from the same program. For example, multiple **EXEC CICS STARTBR** commands can be issued for the same file by the same program. Similarly, a batch program can use multiple request parameter lists (RPLs) for the same ACB. The value you specify for the number of concurrent threads option defines the maximum number of RPLs within all open ACBs that can be used for this data set. For each concurrent thread, the size of the DDM driver module increases, so only specify a value greater than 1 when necessary.

If a program tries to open an ACB and the DIM cannot support another concurrent thread, an abend occurs. This results in a transaction abend code 3105 for a CICS program. A batch program terminates with a U3105 abend code.

Specifying mandatory defaults panel 2

The second defaults panel is for read-only DDMs. All fields are mandatory. If you do not generate read-only DDMs, specify the same values that you use in defaults panel 1.

The fields are shown in Figure 14.

```
 ---------------------- CICS VT: Defaults panel 2 of 4 ----------------------
Command ===> ____________________________________________________________

 Enter defaults for the generation of the CICS VT drivers:-

 Output load library for
 Read Only CICS VT drivers  ==> appl.drivers.load2_____________________

 Output DBRM Library for
 Read Only CICS VT drivers  ==> appl.drivers.dbrm2_____________________

 BIND model in SVIDSAMP for
 Read Only CICS VT drivers  ==> readonly________________________________









 Enter=Confirm   PF1=Help   PF3=Exit
```

Figure 14. CICS VT defaults panel 2 of 4

The field descriptions are as follows:

Table 6. Field descriptions

Field	Description
Output load library for Read Only CICS VT drivers	The name of the load library for read-only DDMs. You can use the same library that you created during configuration and setup for the CICS VT driver modules for each migrated data set. See Figure 3 on page 9. Alternatively, you can generate read-only DDMs into a separate library that you must create yourself.
Output DBRM Library for Read Only CICS VT drivers	The name of the DBRM library for read-only DDMs. You can use the same library that you created during configuration and setup for the CICS VT driver module DBRMs for each migrated data set. See Figure 3 on page 9. Alternatively, you can generate read-only DDM DBRMs into a separate library that you must create yourself.
BIND model in SVIDSAMP for Read Only CICS VT drivers	The name of the read-only DDM skeleton bind member in your *my* .SVIDSAMP. *custom* data set. This member contains default DB2 values that are used during the binding of the CICS VT read-only DDM packages and plans. See "DB2 considerations" on page 58.

The default values you specify apply to all of the read-only drivers that you generate. The **DB2 owner for driver bind** and **No of concurrent threads** options for read-only DDMs are those you specify in the first defaults panel. These can be changed at an individual DIM level, as shown in Figure 35 on page 55.

Specifying mandatory defaults panel 3

The default values in the third defaults panel are for the automated mapping facility and relate specifically to DB2. The panel is shown in Figure 15 on page 36.

```
---------------------------- CICS VT: Defaults panel 3 of 4 ----------------------



Auto mapping:- mandatory defaults

CREATE TABLESPACE <DIM name>
  USING STOGROUP  vidsg__
        SEGSIZE  4_________  (multiple of 4)
     BUFFERPOOL  BPx___


CREATE UNIQUE INDEX <IX/IY_DIM name>
  ON <DIM name>
  USING STOGROUP  VIDSG__
     BUFFERPOOL  BPx___









Enter=Confirm   PF1=Help   PF3=Exit
```

Figure 15. CICS VT defaults panel 3 of 4

The panel values are as follows:

Table 7. Panel values

Field	Description
TABLESPACE USING STOGROUP	Specify the name of the DB2 storage group for the tablespaces.
TABLESPACE SEGSIZE	Specify the default segment size for the segmented tablespaces.
TABLESPACE BUFFERPOOL	Specify the default buffer pool for the tablespaces.
INDEX USING STOGROUP	Specify the name of the DB2 storage group for the index page sets.
INDEX BUFFERPOOL	Specify the default buffer pool for the index page sets.

The default parameters in this panel are mandatory and can be overridden during the automated mapping process.

Specifying optional defaults panel 4

The final mapping default panel contains additional values for creating the DB2 objects. This is shown in Figure 16 on page 37.

```
 ---------------------- CICS VT: Defaults panel 4 of 4 ----------------------
Command ===> ____________________________________________________________________

 Auto mapping:- optional defaults

 Assign value to CURRENT SQLID => sqlid___

 Assign HLQ of CICS VT data sets generated => tso_prefix_

 CREATE TABLESPACE <DIM name>
   LOCKSIZE  ANY
      CLOSE
   COMPRESS  NO
   FREEPAGE
    PCTFREE
 CREATE UNIQUE INDEX <IX/IY_DIM name>
   ON <DIM name>
      CLOSE  NO
      DEFER  NO
   FREEPAGE
    PCTFREE

 Enter=Confirm   PF1=Help   PF3=Exit
```

Figure 16. CICS VT defaults panel 4 of 4

If you supply a value for the field **Assign value to CURRENT SQLID**, the first SQL statement that is executed by the batch job is `SET CURRENT SQLID='your-value'`.

The value you specify for the field **Assign HLQ of CICS VT data sets generated** is the prefix for the data sets generated by the automated mapping facility, such as the data set containing the DB2 DDL. Your TSO prefix is used if this field is blank.

All other fields are optional. DB2 defaults will apply for all fields that you leave blank. Press Enter to return to the CICS VT main menu.

Mapping a KSDS data set

The manual mapping process for a KSDS data set is started by selecting option 2 from the CICS VT main menu. The following screen is displayed:

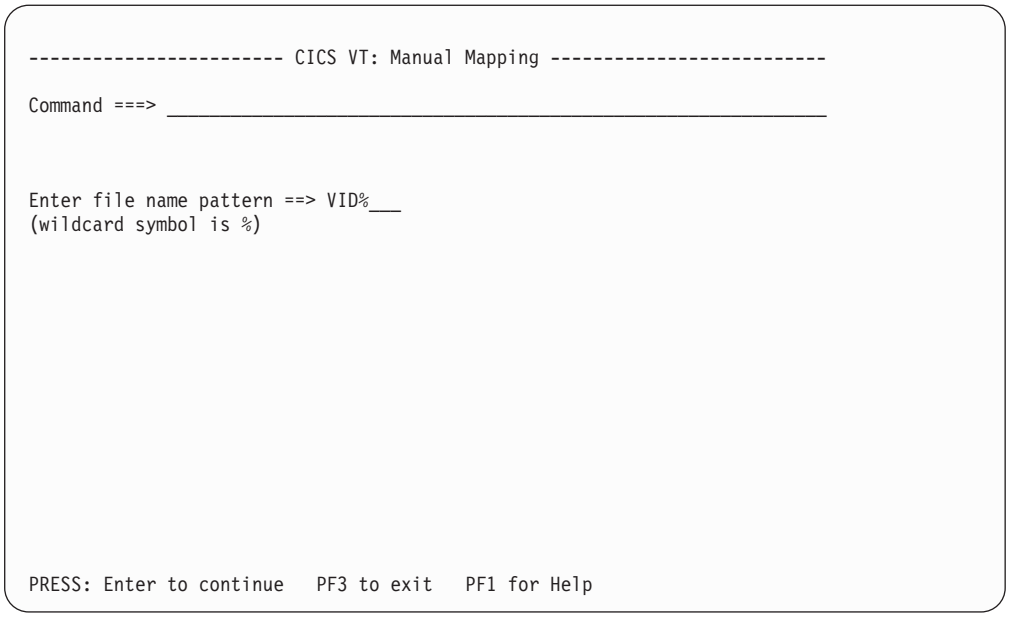


Figure 17. Manual mapping panel

CICS VT uses this panel to build a list of DIMs for further processing. If you are going to work with a file for the first time, specify just the wildcard symbol %.

Adding a new data set to CICS VT

CICS VT builds and displays a list of files that match the DIM name pattern you specify in the screen shown in Figure 17. When you are working with a new data set, specify a blank name pattern and the "Data set list" screen is displayed:

```
------------------------------ CICS VT: Data set list --------------------------
Command ===> ADD_____________________________________________ Scroll ===> CSR_

Commands:      ADD   - Add data set information     ALL   – Generate all

Action codes:  A     - Add data set information
               U     - Update data set information
               S     - Data set mapping information
               G     - Generate CICS VT data set modules (DIM and DDM)
               D     - Delete Data set information from CICS VT

Press: Enter to continue   PF3 to exit   PF1 for help

Actn                                 I/R/D  Ready
code DIM name Build exit I/R/D exit order  for gen Status
---- -------- ---------- ---------- ------ ------- --------------------
  _  ACCOUNT                                      Y
  _  ACCTFIL                                      Y
  _  ACCX001                                      Y
  _  ACC01                                        Y
  _  ACC02                                        Y
  _  ACC03                                        Y
  _  ALTNUNQ                                      Y
  _  AMBSRL                                       Y
  _  APPLCTL             MULTIRD      A      L     Y
  _  AWAAFES                                      Y
  _  AWLONGC1                                     Y
  _  AWMIN                                        Y
  _  AWMSRL                                       Y
  _  AWSQC                                        Y
  _  AWTLRC                                       Y
  _  AWVIDIVP                                     Y
```

Figure 18. CICS VT data set list panel

You add a new file by specifying the command **ADD** on the command line. The following panel is displayed:

```
 --------------------- CICS VT: Add data set information ---------------------
Command ===> ____________________________________________________________________

DIM name                         ===> VIDKSDS

DB2 field build user exit        ===>            If selected, you must build
                                                 all fields for the data set
                                                 with this exit.

DB2 I/R/D exit                   ===>
I/R/D user exit processing order ===>           (B=Before,A=After,' '=Both)
I/R/D user exit language         ===>           (A=Assembler,L=LE enabled)

DIM ready to be generated        ===> N









Enter=Update  PF3=Exit
```

Figure 19. CICS VT add data set information panel

The field descriptions are as follows:

Table 8. Field descriptions

Field	Description
DIM name	A name of 7 characters or less that is unique within CICS VT and represents the VSAM base cluster. For example, you could use the name of the DD that is used to access the base cluster in a batch program. Alternatively, you could use a variation of the CICS file control name and change a single character.
DB2 field build user exit	Leave this field blank unless you are writing an FBE that builds every field in the data set. The *CICS VT Data Reengineering and Customization Guide* deals with exits.
DB2 I/R/D exit	If you are using an IRD exit for data re-engineering, specify the name here. Otherwise leave this field blank.
I/R/D exit processing order	When you use an IRD exit, you must specify whether it should be processed before, after, or both before and after the SQL update call performed by CICS VT. Leave blank unless you specify an IRD exit program.
I/R/D exit language	Specify a value of `A` if the exit is written in assembler, or `L` if it is written in an LE/370 language. Leave blank unless you specify an IRD exit program.
DIM ready to be generated	CICS VT does not allow you to generate a DIM unless this field is set to `Y`. This field value should be specified as `N` until you have completed the mapping for this data set.

The action codes in Figure 18 on page 39 apply to data sets that have already been mapped to CICS VT. This is shown in the next screen:

```
----------------------------- CICS VT: Data set list ----------------------Row 1 of 5
 Command ===> _____________________________________________ Scroll ===> CSR_

 Commands:      ADD   - Add data set information      ALL   – Generate all

 Action codes:  A     - Add data set information
                U     - Update data set information
                S     - Data set mapping information
                G     - Generate CICS VT data set modules (DIM and DDM)
                D     - Delete data set information from CICS VT

 Press: Enter to continue   PF3 to exit   PF1 for help

 Actn                                 I/R/D  Ready
 code DIM name Build exit I/R/D exit order  for gen Status
 ---- -------- ---------- ---------- ------ ------- --------------------
  _   VIDKSDS                              N
  _   VIDK002                              Y
  _   VIDK003                              Y
  _   VIDX001                              Y
  _   VIDX002                              Y
****************************** Bottom of data ********************************
```

Figure 20. CICS VT data set list panel

Action codes are specified at a DSN or DIM level and have the following meanings:

A Add a new data set This is the same as the ADD command.

U Update information for an existing data set. The options you can update are those shown in "Adding a new data set to CICS VT" on page 38.

S Select the data set for mapping.

G Generate a DIM driver. This option is only allowed when the **Ready for gen** field is set to Y.

D Delete all information from CICS VT for this data set.

Note the entry for the DIM name VIDKSDS, which was added in "Adding a new data set to CICS VT" on page 38 earlier as a new file. The **Ready for gen** field for this DIM is set to N, to indicate that the mapping is not complete.

Specifying data set details

After adding the new data set VIDKSDS to CICS VT, select it for mapping by specifying S on the command line. For a new data set the following screen is shown:

```
------------------- CICS VT: Data set mapping for VIDKSDS  -------- Row 1 of 1
Command ===> ________________________________________________ Scroll ===> CSR_

Commands: I - Insert
Actions:  U - Update
          F - Show field information
          D - Delete

Act File type                                                 Status
--- --------- ------------------------------------------- --------------------
****************************** Bottom of data ********************************
```

Figure 21. Insert attributes for a new data set

The first time you process a data set, the message `Select the insert command to add details for a new data set` is displayed. Specify `I` on the command line to display the following screen.

```
---------------------- CICS VT: Insert data set information --------------------
 Command ===> _____________________________________________ Scroll ===> CSR_

 DIM name: VIDKSDS


 File information:

   VSAM filetype  .  ===> KSDS              (KSDS,KSDV,RRDS,RRDV)
   Record length ..  ===> 00080             (Maximum in bytes)
   Fill character .  ===> __                ("FF","00",blank)

 CICS VT information:

   DB2 table:             (Look-up available)
     Table creator   ===> VID..................................................... +
     Table name . .  ===> VID_ITEM................................................ +




 Press: Enter=Update  PF3=Exit
```

Figure 22. Insert file details panel

The information that you must specify is as follows:

Table 9. Required information

Field	Description
VSAM filetype	Specify one of the displayed values, depending on the type of file and whether it is fixed or variable length.
Record length	The maximum length of the record. Do not include the length of the RRN prefix for an RRDS or RRDV file type.
Fill character	If there are obsolete fields that you are not going to migrate to DB2 , the value returned by CICS VT for these fields is the fill character.
Table creator	The SQL authorization ID of the DB2 table for this data set. (This is the value of the CREATOR column in SYSIBM.SYSTABLES.) This is a scrollable field.
Table name	The name of the DB2 table for this data set (This is the value of the column NAME in SYSIBM.SYSTABLES). This is a scrollable field.

CICS VT verifies that the DB2 table exists by checking the DB2 catalog. CICS VT also uses this table for column lookup during field mapping.

Mapping fields

After you enter the data set information in the screen shown in Figure 22 on page 42 the following screen is displayed:

```
------------------- CICS VT: Data set mapping for VIDKSDS  -------- Row 1 of 1
Command ===> _____________________________________________ Scroll ===> CSR_

Commands: I - Insert
Actions:  U - Update
          F - Show field information
          D - Delete

Act File type                                          Status
--- --------- ------------------------------------------- -------------------
 _  KSDS      Bytes: 00080
              Creator:    VID                                              +
              Table name: VID_ITEM                                         +
--- --------- ------------------------------------------- -------------------
****************************** Bottom of data ********************************
```

Figure 23. File details panel

Specify one of the following actions:

U Update details for this data set.

F Select field information for this data set.

D Delete CICS VT information for a data set.

Option `U` is typically used to specify the read-only DDM module name. Expect to use option `D` very infrequently. Option `F`, which you will routinely use, allows you to specify field information and this is the next step in the manual mapping process. Option `F` displays the following panel:

```
---------------------- CICS VT: List of fields for VIDKSDS  ----------------------
Command ===> ________________________________________________ Scroll ===> CSR_

VSAM file type : KSDS     Creator   : VID                                    +
Data set length: 00080    Table name: VID_ITEM                               +

Actions: S Display, U Update, I Insert, D Delete

A Field    Bytes Start Type DB2 column name                 Exit     Pic Par
- -------- ----- ----- ---- ------------------------------   -------- --- ---
 ****************************** Bottom of data ********************************
```

Figure 24. List of fields for a file

Initially, there are no fields defined for a new file. To insert field information, specify option `I` on the command line. Insert field mapping information using the following screen:

```
----------------------------- CICS VT: Insert field -----------------------------
 Command ===> _____________________________________________ Scroll ===> CSR_

 DIM name       : VIDKSDS
 Creator        : VID                                                        +
 Table          : VID_ITEM                                                   +
 Data set length: 00080


 Field name . . .  ===> VIDM0001
 Field length . .  ===> 00006               (In bytes)
 Field type . . .  ===> C                   (C,P,U,F,H,B)
 Column name  . .  ===> ITEM_NUMBER..... + (Look-up available)
 Starting position ===> 00001               ("1" = Beginning of data set)
 Picture  . . . .  ===> ____________________  (example HH.XX.SS.NNNNNN)
                                                   (or MMDDYY)
                                                   (or EXITx=exit name)
 Parameters . . .  ===> ____________________  Optional user parameters
 Special function  ===> KEY                 ("KEY", "PTH", "BKY", or blank)
 Mapped from table ===> P                   ("P"=primary, "X"=not mapped)
 Build order  . .  ===> 00001               ("1"=first, "2"=second and so on)

 Press: Enter=Insert  PF3=Exit  PF1=Help
```

Figure 25. Insert field panel

The field values are as follows:

Field	Description	
Field name	A generated 8-byte value that is unique for the data set. You will normally have one CICS VT field per copybook field.	
Field length	The length of the field. Assuming that you are mapping one copybook field to one DB2 column, this is the length of the copybook field.	
Field type	The options are as follows:	
	C	Character or zoned decimal
	P	Signed packed decimal
	U	Unsigned packed decimal
	F	Fullword (4 bytes binary)
	H	Halfword (2 bytes binary)
	B	Two fullwords (8 bytes binary)
Column name	The name of the DB2 column that the field maps to. If wildcard character % is used, CICS VT builds a selectable list of all matching columns in the table from the DB2 catalog. This field is scrollable, although the maximum DB2 column name is 30 characters.	
Starting position	The start position of this field within the VSAM record. CICS VT calculates this based on the position and length of the preceding field.	
Picture	The layout of date or time fields, or the name of the FBE. See “Specifying a picture string” on page 45.	
Parameters	Optional user parameters. See See “Specifying optional parameters” on page 45.	

Field	Description	
Special function	Special function applies to fields in your VSAM file that are either part of the file key or an alternate index key. A value of KEY is automatically generated for the first inserted field, as shown in Figure 25 on page 44. The values PTH and BKY only apply to alternate index mapping, which is covered in "Mapping the alternate index paths" on page 51. Leave the field blank in all other cases.	
Mapped from table	Always specify P unless you are mapping a multiple field key, in which case specify X. This is explained in "Multiple field key" on page 49.	
Build order	A sequence number generated by CICS VT.	

Specifying a picture string

Use the **Picture** field to provide field-level re-engineering information to CICS VT in two specific cases:

- If the DB2 column you specify in **Column name** has a data type of DATE, TIME, or TIMESTAMP. For example, if the DB2 column is a data type of DATE, you could specify CCYYMMDD as the picture layout for this field.
- If the field will be built by a user written Assembler exit (EXITA=) or an LE/370 language exit (EXITL=).

These cases are discussed in greater detail in the *CICS VT Data Reengineering and Customization Guide*. Leave the **Picture** field value blank if neither case applies.

Specifying optional parameters

You can use nullable DB2 columns but only for columns that do not map to either the base cluster or alternate index fields. Using the optional **Parameters** field, you can specify up to three different single-byte values. If each byte in a VSAM field matches one of the specified values, CICS VT sets the column value to NULL.

Up to three parameters can be specified as follows:

```
NULL[[=input1[input2][input3]],[output],[nullif]
```

Where:

input1-3
: Up to 3 characters that define the repeating input VSAM field value that becomes NULL in DB2

output
: A single character that defines the repeating value in the entire VSAM field that is returned to your application program

nullif
: A single character that the VIDLOAD utility writes in the first byte of the field in the output data. Use this with the NULLIF DB2 LOAD utility parameter.

When multiple input values and an output value are specified, the order of the input values is not significant. When multiple input values are specified without an output value, the output value defaults to the first input value.

Valid input and output values are:

L low-values (x'00's)

H high-values (x'FF's)

S spaces (x'40's)

Z zeroes (x'F0's for zoned decimal or x'0C' or x'0F' for packed decimal)

9 nines (x'F9's for zoned decimal or x'9C' or x'9F' for packed decimal)

Any displayable character is valid for the **nullif** parameter. The value you specify must never be the value of the first byte in a valid field that is mapped to a nullable column.

Example 1

```
NULL
```

In example 1, an input field of low-values becomes a null value in DB2. The value returned from DB2 is low-values, and the **nullif** character in the DB2 load file is "?".

Example 2

```
NULL=S
```

An input field of spaces becomes a null value in DB2. The value returned from DB2 is spaces, and the **nullif** character in the DB2 load file is "?".

Example 3

```
NULL=LHS,S,£
```

An input field of low-values, high-values or spaces becomes a null value in DB2. The value returned from DB2 is spaces, and the **nullif** character in the DB2 load file is "£".

Example 4

```
NULL=,,@
```

An input field of low-values becomes a null value in DB2. The value returned from DB2 is low-values, and the **nullif** character in the DB2 load file is "@".

Example 5

```
NULL=,S
```

An input field of low-values becomes a null value in DB2. The value returned from DB2 is spaces, and the **nullif** character in the DB2 load file is "?".

Be aware of the following before choosing to use nullable columns and the NULLS parameters:

1. Ensure that the **nullif** parameter value you use cannot be the first character of a field that is mapped to a nullable column.
2. Only use `Z` (zeros) and `N` (nines) for fields with types of `C`, `P`, or `U`. These parameter values have no relevance for the other field types.

3. If a field potentially has multiple values that become nulls, the dual mode facility will erroneously signal errors during application testing. Some manual data cleansing in VSAM to eliminate multiple nullable values in the same field is recommended.

When you use the NULLS facility, it is recommended that you also use the DB2 sample REXX DRAW to generate the LOAD utility control cards, instead of the DB2 sample unload program DSNTIAUL.

Mapping an RRDS data set

Unlike a KSDS data set, an RRDS data set does not have a unique key that is part of the data set record. Instead, it uses a fullword relative record number (RRN) to uniquely identify each record.

There are two areas where converting an RRDS data set with CICS VT is different to converting a KSDS data set. One concerns the DB2 primary key, and the other concerns how the data set is mapped. These differences are discussed in this section.

Note that you must be aware of the differences if you are manually mapping a RRDS data set. The automated mapping facility handles the differences for you.

DB2 primary key for an RRDS data set

The DB2 table definition that is used for a migrated RRDS data set is shown in Figure 26 . It uses the sample copybook shown in Figure 9 on page 26 except that in this case the VSAM data set is an RRDS:

```
CREATE TABLE VID_ITEM
  (RRN_NUMBER            INTEGER
  ,ITEM_NUMBER                CHAR(6)
  ,ITEM_NAME                  CHAR(12)
  ,ITEM_COLOUR                CHAR(6)
  ,ITEM_WEIGHT                CHAR(4)
  ,ITEM_COST                  DEC(7,2)
  ,ITEM_REORDER_NO            SMALLINT
  ,ITEM_SUPP_CODE             CHAR(3)
  ,ITEM_DATE_FSHIP            DATE
  ,ITEM_SHELF_LIFE            CHAR(2)
  ,ITEM_DESCRIPTION           CHAR(35)
  (PRIMARY KEY(RRN_NUMBER))
   IN ITEMDB.ITEMTS;
```

Figure 26. DB2 DDL for an RRDS data set table

The column RRN_NUMBER is defined as the DB2 primary key. The name of this column is optional, but to map an RRDS data set you must have an INTEGER column as the primary key of the DB2 table.

Mapping differences for an RRDS data set

There are two differences in the mapping process for an RRDS data set. The first difference is in the **VSAM file type** field, which must have a value of RRDS or RRDV. This is specified in the "Insert file panel", which is shown in Figure 22 on page 42.

The second difference concerns the mapping of the key field, as illustrated in Figure 27 on page 48.

```
---------------------- CICS VT: List of fields for VIDRRDS  ------- Row 1 of 11
Command ===> ____________________________________________ Scroll ===> CSR_

VSAM file type : RRDS     Creator   : VID                                    +
Data set length: 00080    Table name: VID_ITEM                               +

Actions: S Display, U Update, I Insert, D Delete

A Field    Bytes Start Type DB2 column name                  Exit     Pic Par
- -------- ----- ----- ---- ------------------------------   -------- --- ---
_ RRDSKEY  00001 00001   X                                 +
_ VIDF0001 00006 00001   C  ITEM_NUMBER                    +
_ VIDF0002 00012 00007   C  ITEM_NAME                      +
_ VIDF0003 00006 00019   C  ITEM_COLOUR                    +
_ VIDF0004 00004 00025   C  ITEM_WEIGHT                    +
_ VIDF0005 00004 00029   P  ITEM_COST                      +
_ VIDF0006 00003 00033   C  ITEM_REORDER_NO                +
_ VIDF0007 00003 00036   C  ITEM_SUPP_CODE                 +
_ VIDF0008 00005 00039   P  ITEM_DATE_FSHIP                +           Y
_ VIDF0009 00002 00044   C  ITEM_SHELF_LIFE                +
_ VIDF0010 00035 00046   C  ITEM_DESCRIPTION               +
```

Figure 27. Data set mapping for an RRDS data set

The first field RRDSKEY does not map to a DB2 column. You must update the RRDSKEY field by selecting action U. This shows the following screen:

```
----------------------------- CICS VT: Update field -----------------------------
 Command ===> ____________________________________________ Scroll ===> CSR_

 DIM name       : VIDKSDS
 Creator        : VID                                                          +
 Table          : VID_ITEM                                                     +
 Data set length: 00080

 Field name . . . . . : RRDSKEY
 Field length . . .  ===> 00001             (In bytes)
 Field type . . . .  ===> X                 (C,P,U,F,H,B)
 Column name  . . .  ===> ........................... + (Look-up available)
 Starting position   ===> 00001             ("1" = Beginning of data set)
 Picture or FBE . .  ===> ____________________________  (example  HH.XX.SS.NNNNNN)
                                                        (or MMDDYY)
                                                        (or EXITx=exit name)
 Parameters . . . .  ===> ____________________________  Optional user parameters
 Special function .  ===> KEY               ("KEY", "PTH", "BKY", or blank)
 Mapped from table   ===> X                 ("P"=Prim, "X"=Not mapped)
 Build order  . . .  ===> 00001             ("1"=first, "2"=second and so on)


 Press: Enter=Update  PF3=Exit  PF1=Help
```

Figure 28. Updating the key field for an RRDS data set

Specify KEY in the **Special function** field and X in the **Mapped from table** field.

When you map an RRDS data set manually, you must specify this field exactly as it is shown in Figure 28.

The mapping of the copybook fields for an RRDS data set is the same as the field mapping for a KSDS data set.

CICS VT mapping rules

There are some mapping rules that CICS VT imposes, which apply to mapping all supported data set types. The automated mapping feature applies these rules, but you must apply them when using the manual mapping method.

Figure 29 shows an example of the complete mapping data for the file VIDKSDS. The copybook and DDL for this file are shown in "The mapping component" on page 26.

```
---------------------- CICS VT: List of fields for VIDKSDS  ------- Row 1 of 10
Command ===> _____________________________________________ Scroll ===> CSR_

VSAM file type : KSDS     Creator   : VID                                  +
Data set length: 00080    Table name: VID_ITEM                             +

Actions: S Display, U Update, I Insert, D Delete

A Field    Bytes Start Type DB2 column name                  Exit     Pic Par
- -------- ----- ----- ---- ------------------------------   -------- --- ---
_ VIDKEY   00006 00001   C  ITEM_NUMBER                     +
_ VIDM0002 00012 00007   C  ITEM_NAME                       +
_ VIDM0003 00006 00019   C  ITEM_COLOUR                     +
_ VIDM0004 00004 00025   C  ITEM_WEIGHT                     +
_ VIDM0005 00004 00029   P  ITEM_COST                       +
_ VIDM0006 00003 00033   C  ITEM_REORDER_NO                 +
_ VIDM0007 00003 00036   C  ITEM_SUPP_CODE                  +
_ VIDM0008 00005 00039   P  ITEM_DATE_FSHIP                 +           Y
_ VIDM0009 00002 00044   C  ITEM_SHELF_LIFE                 +
_ VIDM0010 00035 00046   C  ITEM_DESCRIPTION                +
```

Figure 29. List of fields panel

The mapping rules imposed by CICS VT are as follows:

- Each field must have a unique name, as shown in the **Field** field in Figure 29. The name must be between 1 and 8 characters long. The field names in Figure 29 are generated by the CICS VT automated mapping feature.
- The first field must always have the function KEY.
- Two DB2 columns cannot map to the same CICS VT field.

Note that the **Bytes**, **Start**, and **Type** fields refer to the attributes of the copybook fields, not the DB2 column. For example, the copybook field ITEM-REORDER-QUANTITY, which is shown in Figure 9 on page 26, has an attribute of PIC S999, which is 3 bytes. The field name generated by CICS is VIDM0006 and is mapped to column ITEM_REORDER_NO, which is SMALLINT with a length of 2 bytes in DB2.

Multiple field key

The first field that you map for a data set must always be the key field. In cases where the key is made up of multiple copybook fields, the mapping must follow specific rules. For example, assume that you have a KSDS data set with a 12-byte key field that is broken down in the copybook to two 6-byte fields. The mapping for this file is shown in Figure 30 on page 50:

```
---------------------- CICS VT: List of fields for MULTKEY  ------- Row 1 of 4
Command ===> _____________________________________________ Scroll ===> CSR_

VSAM file type : KSDS     Creator   : VID                                    +
Data set length: 00080    Table name: VID_MULTKEY                          +

Actions: S Display, U Update, I Insert, D Delete

A Field    Bytes Start Type DB2 column name                 Exit     Pic Par
- -------- ----- ----- ---- ------------------------------  -------- --- ---
_ GROUPKEY 00012 00001   C                                +
_ KEYFLD1  00008 00001   C  KEY_COL_01                     +
_ KEYFLD2  00004 00009   C  KEY_COL_02                     +
_ VIDF0001 00015 00013   C  FIRST_DATA_COL                 +
```

Figure 30. Mapping a multiple field key

The field called GROUPKEY represents the composite key field, starting in position 1 for a length of 12 bytes. Because this field does not map exactly to a single DB2 column, it is designated as unmapped. This means that it is simply a placeholder, and the DB2 column name for this field is blank.

You must select option U for the GROUPKEY field. The following screen is displayed:

```
----------------------------- CICS VT: Update field -----------------------------
 Command ===> _____________________________________________ Scroll ===> CSR_

 DIM name       : MULTKEY
 Creator        : VID                                                          +
 Table          : VID_MULTKEY                                                  +
 Data set length: 00080

 Field name . . . . . . : GROUPKEY
 Field length . . .  ===> 00012             (In bytes)
 Field type . . . .  ===> C                 (C,P,U,F,H,B)
 Column name  . . .  ===> ........................... + (Look-up available)
 Starting position   ===> 00001             ("1" = Beginning of data set)
 Picture or FBE . .  ===> ____________________________  (example  HH.XX.SS.NNNNNN)
                                                        (or MMDDYY)
                                                        (or EXITx=exit name)
 Parameters . . . .  ===> ____________________________  Optional user parameters
 Special function .  ===> KEY               ("KEY", "PTH", "BKY", or blank)
 Mapped from table   ===> X                 ("P"=Prim, "X"=Not mapped)
 Build order  . . .  ===> 00001             ("1"=first, "2"=second and so on)


 Press: Enter=Update  PF3=Exit  PF1=Help
```

Figure 31. Updating the field for a group key field

Specify KEY in the **Special function** field and X in the **Mapped from table** field.

Immediately following the GROUPKEY fields in Figure 30 are two copybook fields that are mapped to two DB2 columns KEY_COL_01 and KEY_COL_02. These are the primary key columns of the DB2 table. The first field is called KEYFLD1. The panel shows that it starts at position 1 and has a length of 8 bytes. The second part of the key is called KEYFLD2, which starts at position 9 and has a length of 4 bytes.

Mapping alternate indexes

VSAM uses alternate indexes to provide a different path through the data set. In CICS VT, every alternate index path must be mapped. If you are using the automated mapping facility, alternate indexes are mapped automatically. If you are using the manual mapping facility, you must map alternate indexes manually. A CICS VT batch utility program is provided to map alternate indexes manually and this section describes what you need to do.

Identifying alternate index paths

The first step in mapping an alternate index is to identify the characteristics of the path. Use the access method services IDCAMS program to list the integrated catalog facility (ICF) cluster entry for the alternate index. The sample IDCAMS input statement that follows is an alternate index cluster that provides a path to the KSDS data set from earlier in this chapter in Figure 8 on page 26.

```
LISTCAT ENT(VID.VIDKSDS.AIX) ALL
```

In the IDCAMS output listing, look for the following fields:

```
ATTRIBUTES
 KEYLEN-----------------3
 RKP--------------------5
 AXRKP------------------35
```

The KEYLEN and AXRKP fields are significant. KEYLEN is the length of the field, and AXRKP is the offset of the alternate index field in the record, relative to zero. In this example, the key field is 3 bytes starting at the 36th byte.

If you see the attribute NONUNIQKEY, additional mapping is required. This is described in "Non-unique alternate indexes" on page 53.

In the mapping of the VIDKSDS data set earlier in this chapter, the field starting at byte 36 for a length of 3 bytes is called VIDF0006. You need this information in the next step.

Note that when you map a data set base cluster that has alternate indexes, you must ensure that alternate index fields map to one or more complete DB2 columns.

Mapping the alternate index paths

The next step is to map the path to CICS VT using a supplied batch utility. This step applies to both unique and non-unique indexes. Every alternate index has a unique DIM and DDM. When you manually map an alternate index, you specify a unique DIM name in the VIDPATHJ utility. Sample JCL is provided in member VIDPATHJ of the *my*.SVIDSAMP.*custom* library. The JCL looks like this:

```
//INDEX   EXEC PGM=VIDPATHM
//STEPLIB  DD DSN=VID.SVIDLODE,DISP=SHR
//         DD DSN=DB2.SDSNLOAD,DISP=SHR
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSIN    DD  *
PATHMAP BASENAME=VIDKSDS,PATHNAME=KSDSAIX
PATHKEYS=VIDM0006
```

Figure 32. Sample JCL for CICS VT utility to map an alternate index

The parameters in the SYSIN cards have the following meaning:

Parameter	Description
PATHMAP	This indicates that the SYSIN parameters are to map an alternate index.
BASENAME	The name of the CICS VT DIM that corresponds to the base VSAM cluster.
PATHNAME	The user-specified DIM name that corresponds to the PATH. The length of the name should not exceed 7 characters.
PATHKEYS	The CICS VT field name for the alternate key. If there are multiple fields because the alternate key maps to multiple DB2 columns, separate them using commas. For example, PATHKEYS=VIDM0006,VIDM0007,VIDM0008.

The batch job creates all of the mapping information that CICS VT needs and should end with return code 0000. The following message is produced:

```
PROCESSING COMPLETE FOR PATH pathname
  ON BASE basename
```

You must run the VIDPATHJ utility for every alternate index on a base data set.

Mapping alternate indexes with multiple fields

If the number of fields in the alternate index cannot be specified in a single **PATHKEYS** statement, you must specify the **PATHKEYS** keyword on every subsequent line. Here are two examples for the same alternate index:

Example 1:

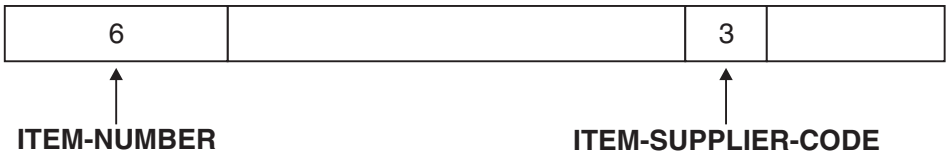
```
PATHMAP BASENAME=mydim,PATHNAME=myaix
PATHKEYS=VIDFL007
PATHKEYS=VIDFL008
PATHKEYS=VIDFL009
PATHKEYS=VIDFL010
PATHKEYS=VIDFL011
PATHKEYS=VIDFL012
PATHKEYS=VIDFL013
PATHKEYS=VIDFL014
PATHKEYS=VIDFL015
PATHKEYS=VIDFL016
PATHKEYS=VIDFL017
PATHKEYS=VIDFL018
PATHKEYS=VIDFL019
```

Example 2:

```
PATHMAP BASENAME=mydim,PATHNAME=myaix
PATHKEYS=VIDFL007,VIDFL008,VIDFL009,VIDFL0010,VIDFL011,VIDFL0012
PATHKEYS=VIDFL013,VIDFL014,VIDFL015,VIDFL0016,VIDFL017,VIDFL0018
PATHKEYS=VIDFL019
```

Creating the appropriate DB2 indexes

When an application program issues a call to a VSAM data set using an alternate index path, the CICS VT DDM SQL is predicated on the columns that correspond to the path key. For example, the layout of the record for the sample file as shown in the copybook is as follows:



To access this file using the alternate index path, CICS VT uses the following predicate:

```
WHERE ITEM_SUPP_CODE =:host variable.
```

You must create a DB2 index on the column or columns that correspond to the alternate key. In this example, the field ITEM-SUPPLIER-CODE is mapped to the single DB2 column ITEM_SUPP_CODE.

Non-unique alternate indexes

Mapping non-unique secondary indexes is slightly different to mapping unique secondary indexes. To ensure that CICS VT retrieves the records in exactly the same sequence as VSAM, you must add a column to the DB2 table.

The column name must be VID_PATH_DUPKEY and the data type must be TIMESTAMP. The column can be in any position in the table after the primary key column or columns and must not be nullable. The VID_PATH_DUPKEY column should be the last column in the DB2index.

The data value in this column is managed by CICS VT. The column must be included in the table prior to generating the drivers and performing data migration.

Generating CICS VT runtime modules

The “Functional overview” on page 25 introduced the two runtime modules used by CICS VT: the DIM and the DDM. These modules are ready to be generated when the mapping of each VSAM data set and its alternate indexes is complete.

The DIM driver is used by CICS VT to understand the relationship between the VSAM record structure and the DB2 row structure. The DDM driver contains the SQL that CICS VT uses to access the migrated data sets in DB2. These modules are generated automatically by the automated mapping facility. When you manually map data sets, you must generate the driver modules using the CICS VT ISPF facility.

Before you generate your first drivers, you should review “DB2 considerations” on page 58 and “The DDM driver module name” on page 59.

Generating the DIM

Driver modules are generated by batch utilities. There are two stages in the generation process. You initiate the driver generation process by selecting option 2 from the CICS VT "Data set list" panel, shown in Figure 33 on page 54:

```
----------------------------- CICS VT: Data set list ----------------------Row 1 of 5
Command ===> _____________________________________________ Scroll ===> CSR_

Commands:      ADD   - Add data set information   ALL      - Generate all

Action codes:  A     - Add data set information
               U     - Update data set information
               S     - Data set mapping information
               G     - Generate CICS VT data set modules (DIM and DDM)
               D     - Delete data set information from CICS VT

Press: Enter to continue   PF3 to exit   PF1 for help

Actn                                 I/R/D  Ready
code DIM name Build exit I/R/D exit order  for gen Status
---- -------- ---------- ---------- ------ ------- --------------------
 G   VIDKSDS                                N
 _   VIDK002                                Y
 _   VIDK003                                Y
 _   VIDX001                                Y
 G   VIDX002                                Y
****************************** Bottom of data ********************************
```

Figure 33. Data set list panel

Specify G in the **Actn code** for the DIM you want to generate. CICS VT will only allow you to select a DIM for generation if the **Ready for gen** field has the value Y. If this value is N, specify a value of U in the **Actn code** field, then specify a value of Y in the DIM **Ready to be generated** field of the "Update data set information" panel, shown in Figure 34.

```
 ---------------------- CICS VT: Update data set information ---------------------
Command ===> ____________________________________________________________________

DIM name                         ===> VIDKSDS

Field build user exit            ===> ________   If selected, you must build
                                              all fields for the dataset
                                              with the specified exit.

I/R/D user exit name             ===>
I/R/D user exit processing order ===>          (B=Before,A=After,' '=Both)

DIM ready to be generated        ===> Y









Enter=Update  PF3=Exit
```

Figure 34. Update data set information panel

You can generate drivers for more than one data set or alternate index at the same time.

When you have made your selection, CICS VT redisplays this panel showing the DIMs that you have selected. Press Enter to continue and the job submit panel is presented, as shown in Figure 35 on page 55.

```
------------------------ CICS VT: Generate drivers ------------------------------
 Command ===> ____________________________________________________________________


 DB2 owner for driver bind    ===> DBADM___

 No of concurrent threads     ===> 01         01-99   PF1 for more information

 Automatically submit stage 2 ===> Y

 SUBMIT or EDIT generated JCL ===> SUBMIT

 Generated JCL data set name  ===> ____________________________________________
 (If not specified then data set 'prefix.VID.ISPFILE' will be allocated)

 Verify job card statement:
 ===> //VIDJOB JOB (ACCT,PGMer-name),CLASS=L,MSGCLASS=X,NOTIFY=&SYSUID__ <===
 ===> //WHERESMY JCLLIB ORDER=my.SVIDCNFG.custom_____________________________ <===
 ===> ____________________________________________________________________ <===
 ===> ____________________________________________________________________ <===


 Press: Enter to continue   PF3 to exit   PF1 for Help
```

Figure 35. Generate DIM and DDM job submit panel

The field descriptions in this panel are as follows:

Field	Description
DB2 owner for driver bind	The value of the **OWNER** parameter for the DB2 package and plan bind. This overrides the default value in Figure 13 on page 33.
No of concurrent threads	The maximum number of concurrent threads per user for this data set. This overrides the default value in Figure 13 on page 33.
Automatically submit stage 2	There are two separate jobs to generate CICS VT drivers, referred to as generation stage 1 and generation stage 2. If you specify a value of `Y` for the **Automatically submit stage 2** option, the second job is automatically submitted. A value of `N` means that you have to submit it manually. See "Driver generation stage 2" on page 56 for more information.
SUBMIT or EDIT generated JCL	Specify `SUBMIT` to submit the stage 1 generation job automatically. If you specify `EDIT` you can review or edit the generated JCL, in which case you must manually submit the job for processing. The `SUBMIT` and `EDIT` commands can be abbreviated to `S` and `E`.
Generated JCL data set name	Specify a sequential data set where the JCL will be created. This is a temporary data set that will be created if it doesn't exist. Your TSO profile prefix is used as the high-level qualifier for the temporary data set. If you don't use a TSO profile prefix, the value you specified for the field **Assign HLQ of CICS VT data sets generated** in Figure 16 on page 37 is used.

Field	Description
Verify job card statement	The job card that is used for the generation jobs.

Driver generation stage 1

There are five steps in the generation stage 1 batch job and they all should end with a return code of 0. The steps are as follows:

Table 10. Generation stage 1 batch job steps

Step #	Step Name	Function
1	GEN	Generate the DIM source code
2	ASM	Assemble the generated code.
3	LINK	Link the object code into an executable module
4	CLEANUP	Delete temporary data sets
5	VIDDDMG	Generate the job deck for stage 2

The GEN step verifies the that mapping information is correct by performing the following checks:

- Ensuring that the DB2 table exists.
- Checking that the column names specified in the mapping exist in the table.
- Checking that the mapping conforms to the rules imposed by CICS VT.
- Ensuring that the combination of the field data type and the DB2 column data type is supported by CICS VT. For example, a field defined as *packed*, that is mapped to a DB2 column with a data type CHAR, is an unsupported conversion, and will result in an error in the GEN step.

There are two outputs from the stage 1 generation job. A DIM module is created in the CICS VT drivers library that you specified in the first defaults panel and the job deck is generated for the stage 2 batch job.

VID_SRC control table

The VIDDDMG step in the driver generation stage 1 job inserts records in the VID_SRC control table. These DB2 records are used in the driver generation stage 2 job. After the generation stage 2 job has completed, the data in the VID_SRC table is no longer required.

You might encounter a DB2 -904 SQL code in the VIDDDMG step if you are generating multiple drivers at the same time and the tablespace VIDTS4 is unable to extend. If this occurs, use SPUFI to delete all the records in this table and rerun the generation stage 2 job. If the problem still persists, increase the space allocation for the VIDTS4 tablespace, run the DB2 LOAD utility with a null input data set and the option REPLACE, then rerun the generation stage 2 job.

Driver generation stage 2

If you specified a value of `Y` for the **Automatically submit stage 2** option, as shown in Figure 35 on page 55, the second part of the driver generation is

automatically submitted. If you specified a value of N, you must manually submit the stage 2 batch job. The JCL for this is created by the stage 1 batch job in a sequential data set called *tsoprefix*.VID.STAGEII. No changes to the JCL are required. The steps in the stage 2 job are as follows:

Table 11. Driver generation stage 2 steps

Step #	Step Name	Function
1	SQ000000	Extract data generated in stage 1 from DB2.
2	PC	Precompile with the DB2 DSNHPC program.
3	SPLIT	Split the modified source from the PC step.
4	ASMB001	Assemble the first CSECT.
5	ASMB002	Assemble the second CSECT.
6	LINK	Link the object decks into an executable module.
7	BIND	Bind the DBRM from the precompile step.

The PC step always ends with a return code of 04, and this can be ignored. The LINK step might end with a return code of 04, depending on local system defaults. You can ignore this. All other steps should return a condition code of zero.

The number of assembly steps (ASMB00x) depends on the value specified for the **No of concurrent threads** parameter. A value of 1 results in two assembly steps, a value of 3 results in six assembly steps, and so on.

The output from the generation stage 2 job is a DDM driver and a DB2 DBRM.

Requesting a read-only DDM

The default DDM contains SQL to support all the possible calls you can make to a VSAM data set. In certain circumstances, you may want to restrict access to prevent data updates, and CICS VT provides the ability to generate a read-only DDM facility for this purpose.

You request a read-only DDM in the manual mapping dialogs. Select the DIM and specify U to update the DIM mapping. This display the screen shown in Figure 36 on page 58:

```
---------------------- CICS VT: Update data set mapping ------------------------
 Command ===> ____________________________________________________ Scroll ===> CSR_

 DIM name        :dim-name


 File information:

   VSAM filetype  .  ===> KSDS               (KSDS,KSDV,RRDS,RRDV)
   Record length ..  ===> 04096              (Maximum in bytes)
   Fill character .  ===> __                 ("FF","00",blank)

 CICS VT information:

   DB2 table:             (Look-up available)
     Table creator   ===> DBADM..................................................... +
     Table name . .  ===> DB2-table................................................. +


 Read Only DDM       ===> module________           (Optional)


 Press: Enter=Update  PF3=Exit
```

Figure 36. Data set mapping

Specify the name of the read-only DDM you want to generate. You can specify any valid name.

The DB2 bind parameters for the read-only DDM come from the skeleton bind member that you specify in the mapping defaults panel 3, shown in “Specifying mandatory defaults panel 2” on page 34. The read-only DDM is automatically generated when you generate the DIM and read-write DDM.

See “VSAM file DD statement changes” on page 97 for the JCL changes that are required to use the read-only DDM at run time.

DB2 considerations

A DDM contains SQL and produces a DB2 DBRM that is bound into a DB2 package in the step called BIND. The parameters that CICS VT uses in this step are provided by a skeleton that you can customize.

The skeleton is a member in the *my* .SVIDSAMP. *custom* data set, and the default name is VIDBIND. Customize this member so that the DB2 parameters for the DDM conform to your local DB2 standards. Tailoring instructions are included as comments in the default member.

CICS VT requires the default DB2 plan to match the DIM name. You can override the plan name using a JCL DD statement, which is explained in “ Overriding the default DB2 plan name ” on page 128 .

Instead of updating the VIDBIND member, you can create one or more alternative members in *my* .SVIDSAMP. *custom* . Specify the name of the member you want to use in the CICS VT main menu option *BIND model in SVIDSAMP* . For example, your DB2 local standards might dictate that you have different collections for different migrated data sets. In this case, create multiple bind skeletons, and specify which one to use in the panel shown in Figure 35 on page 55 . You should also consider creating a bind skeleton with the EXPLAIN (YES) option in order to verify that your DB2 access paths are correct. The PLAN_TABLE should show a matching index scan for all CICS VT access paths.

If your application program already contains SQL, then you will have existing DB2 packages bound into an existing plan. In this case, rebind the existing application plan to include both the existing packages and the CICS VT DDM driver package. You should understand clearly the operational implications of migrating a VSAM data set to DB2 before you migrate your first file. These are detailed in Chapter 8, "CICS VT operational and recovery considerations," on page 125 .

The DDM driver module name

The name of the DDM module is derived from the combination of the name of the DIM and a CICS VT exit program called VIDDDMEX. If the DIM name is 7 characters or less, the default VIDDDMEX module supplied in VID.SVIDLODE adds the character '#' to the end of the DIM name. The source and assemble/link JCL for the default VIDDDMEX module is supplied in *my*.SVIDCNFG.*custom* member VIDGDMEX. You can modify the exit to produce a DDM name that is valid in your environment.

The automated mapping facility

When you manually map a data set, you are responsible for creating the DDL, mapping the data set with the ISPF mapping dialogs, and generating the driver modules. Alternate indexes must be mapped manually. All of these actions are performed by the automated mapping facility, including the generation of the DB2 DDL and CICS VT driver modules for alternate indexes associated with a base cluster. You perform the following steps with the automated mapping facility:

1. Gather information about the data set or data sets to be mapped.
2. Select **Auto mapping facility** from the CICS VT main menu, and the DIM name to be processed using the CICS VT ISPF dialogs.
3. Specify the copybook you want to use to map this DIM.
4. Optionally edit the mapping data. For example, you might want to change a DB2 column name or attribute. CICS VT always presents the edit panel if a user action is required; for example, if there are duplicate field names in the copybook. Mapping data is inserted into the CICS VT control tables when editing is complete.
5. Submit the batch auto mapping generation job. This creates the DB2 objects and generates the drivers.
6. Perform post auto-mapping tasks.

The rules and detailed information on how to perform these tasks are explained in the following section.

Rules for using automated mapping

You can use the automated mapping facility in the following circumstances:

- There is a valid copybook for a data set and it will successfully compile.
- The data set will migrate to a single table.
- The input copybook member defines a single record structure and does not contain multiple record structures.
- There is a maximum of five levels of nested redefined fields or group fields.
- Assembler copybooks must not be macros. All MACRO and MEND statements must be removed before the copybook is processed.

Data sets that do not conform to these rules must be mapped using the manual mapping method.

An error message is issued and the mapping is terminated if the total number of columns exceeds the DB2 limit of 750. You must reduce the total number of copybook fields to less than 750 by combining or removing fields. This situation can occur if you have repeating group fields.

Automated mapping is also terminated if the copybook length exceeds the VSAM record length. An appropriate error message is issued.

Note: It is impossible for the automated mapping facility to process every conceivable copybook. You should consider editing complex copybooks before trying to process them with the automated mapping facility.

Gathering mapping information

You can use a batch utility to collect information about one or more VSAM data sets and write it to the CICS VT control tables. The member VIDAUTOJ in VID.SVIDSAMP is sample JCL for this utility, and is shown in Figure 37.

```
//AUTOMAP  EXEC PGM=VIDMAPIN
//STEPLIB  DD  DSN=VID.SVIDLODE,DISP=SHR
//         DD  DSN=DB2.SDSNLOAD,DISP=SHR
//SYSIN    DD  UNIT=VIO,SPACE=(CYL,(1,1))
//SYSPRINT DD  UNIT=VIO,SPACE=(CYL,(1,1))
//SYSOUT   DD  SYSOUT=*
//REPORT   DD  SYSOUT=*
//CATIN    DD  *
VIDKSD1 VID.KSDS01
VIDKSD2 VID.KSDS02
VIDKSD3 VID.KSDS03
```

Figure 37. Sample JCL for VIDAUTOJ utility

The input to this utility is shown after the CATIN DD statement. The first parameter is the DIM name you select for this data set, and the second parameter is the name of the base cluster. The first parameter must start at position 1, and the second parameter must start at position 9. You can change the UNIT=VIO parameter for the DD statements SYSIN and SYSPRINT to UNIT=SYSDA or equivalent if VIO is not defined on your system.

Alternatively, you can use the ISPF dialogs to gather mapping information for a single data set using the `ADD` command shown in "Selecting the DIM to be processed" on page 61. There is an auto-mapping entry for a DIM called DUMDIM, which is added during the installation of CICS VT. This can be deleted after you have added a new DIM.

CICS VT calculates the size of the DB2 table and index objects from the primary and secondary space allocation information stored in the base cluster. If you gather mapping information using a test version of the data set cluster, you might have to manually update the tablespace and indexspace space parameters when you convert the production version of the data set.

The batch utility can be rerun if required, in which case previously gathered information for this DIM is deleted. Using the ISPF dialogs, if you attempt to analyze an already analyzed DIM, an error message is issued and no data is updated. If you need to reanalyze a data set, use the ISPF option `F` in the Auto mapping list panel first.

Selecting the DIM to be processed

Use the CICS VT ISPF mapping component to perform the next automated mapping step. Ensure that you have defined the CICS VT mapping defaults before using the automated mapping facility. These are described in "Defining mapping defaults" on page 32.

Select option 1 **Auto mapping facility** from the CICS VT main menu. This displays the automated mapping entry panel, as shown in Figure 38:

```
------------------------- CICS VT: Auto mapping entry panel ----------------------
  Command ===> ____________________________________________________________________


  Selection of DIM names processed by the VIDMAPIN utility:-

  Enter file name pattern ==> %_______
  (wildcard symbol is %)



  Enter job card statement to be used by generated batch jobs:-
  ===> //VIDJOB JOB (ACCT,PGMer-name),CLASS=L,MSGCLASS=X,NOTIFY=&SYSUID __ <===
  ===> //WHERESMY JCLLIB ORDER=my.SVIDCNFG.custom_____________________________ <===
  ===> ____________________________________________________________________ <===
  ===> ____________________________________________________________________ <===


  Press: Enter to continue   PF3 to exit   PF1 for help
```

Figure 38. CICS VT automated mapping entry panel

Specify the DIM name pattern and press Enter. CICS VT builds a list of all DIMs that match the file name pattern and displays them in the following screen:

```
------------------------- CICS VT: Auto mapping list ------------ Row 1 of 11
Command ===> _____________________________________________ Scroll ===> CSR_

Press: Enter to continue   PF3 to exit   PF1 for help

Commands: ADD Add new data set information (use for the first entry)

Actions:  S - Select for automated mapping   D - Delete data set information
          A - Add new data set information   F - Delete all information for DIM

Act DIM name Data set name                                 Status
--- -------- ---------------------------------------------- --------------------
 _   LAAFES   CICSVT.LCN.AAFES                              DIM mapped
 _   LCCTFIL  CICSVT.LCN.ACCTFIL                           DIM mapped
 _   LCNMIN1  CICSVT.LCN.LCNMIN                            DIM mapped
 _   LCNMIN2  CICSVT.LCN.LCNMIN                            DIM mapped
 _   LCNMIN3  CICSVT.LCN.LCNMIN                            DIM mapped
 _   LCNMIN4  CICSVT.LCN.LCNMIN                            Ready to map
 _   LCN750   CICSVT.LCN.LCN750                            DIM mapped
 _   LFBTLRC  CICSVT.LCN.FBTLRC                            Ready to map
 _   LLONGC1  CICSVT.LCN.LONGC1                            Ready to map
 _   LONGC1   CVTLONG.LONGC1                               DIM mapped
 S   VIDKSDS  VID.VIDKSDS                                  Ready to map
```

Figure 39. CICS VT auto mapping list

A data set is only eligible for automated mapping if the status is **Ready to map**. Specify S in the **Act** field for automated mapping.

Option A performs exactly the same as the ADD command, and presents a screen to supply information for a new data set.

If you want to delete a data set that you have mapped previously, shown as DIM mapped in the status field, specify D in the **Act** field. This deletes all of the mapping information and all the DB2 objects previously created for the data set. However, data for this DIM that was gathered by analysing the VSAM cluster is not deleted. Use F to delete the mapping information and previously gathered information. Options D and F display confirmation screens.

Specifying the copybook

The next stage is to specify the data set copybook. When you specify S in Figure 39 on page 61 you see the panel shown in Figure 40.

```
------------------ CICS VT: Auto mapping input for VIDKSDS  -------------------
Command ===> ____________________________________________ Scroll ===> CSR_

Mandatory fields:-
Copybook data set ===> CICSVT.LCN.COPYBOOK_____________________________________
Copybook member   ===> ________ (Blank or pattern for member selection list)
Copybook language ===> C        (Assembler or COBOL or PL/I)
Zoned to DECIMAL  ===> Y        (Y/N, applies to unsigned zoned only)
Review / edit DDL ===> Y        (Y/N)

Table creator     ===> VID..................................................... +
Table name        ===> VID_ITEM_TABLE.......................................... +
Primary index     ===> VID_ITEM_INDEX.......................................... +
Tablespace name   ===> VIDITTS_
Database name     ===> VIDITDB_

Optional fields:-
Copybook first field => _____________________________________________________
Copybook last field  => _____________________________________________________

DDL output data set  => _____________________________________________________

Press: Enter to continue   PF3 to exit   PF1 for help
```

Figure 40. CICS VT auto mapping input panel

The fields in the auto mapping input panel are as follows:

Table 12. Auto mapping input panel fields

Field	Description
Copybook data set	The name of the data set containing the copybook for this data set.
Copybook member	The name of the copybook member. Leave this field blank for a member selection list.
Copybook language	Specify the first letter of the input copybook member language. Note that results are unpredictable if the incorrect language is specified.
Zoned to DECIMAL	Specify Y if you want unsigned zoned decimal fields to be mapped to DB2 DECIMAL columns. N will map the fields to CHAR columns.

Table 12. Auto mapping input panel fields (continued)

Field	Description
Review / edit DDL	`Y` takes you to the Edit columns screen. You can perform various tasks such as renaming or updating columns. `N` bypasses the Edit columns screen and submits the automated mapping batch jobs, but only if no user actions are required.
Table creator	Specify the creator of the DB2 table and indexes. This field is scrollable.
Table name	Specify the name of the DB2 table. This field is scrollable.
Primary index	Specify the name of the primary index for the DB2 table. This field is scrollable.
Tablespace name	The DB2 tablespace for the table. This defaults to the DIM name.
Database name	The DB2 database for the tablespace. The database must already exist.
Copybook first field	If the input copybook member contains multiple record layouts, specify the name of the first field (without the field level number).
Copybook last field	Specify the last field name in the structure.
DDL output data set	Specify the name of the sequential data set for the DDL that CICS VT will generate for the DB2 objects. This data set should be saved in order to transfer the file to a different system. If the data set exists it is overwritten.

CICS VT does not allow mapping to proceed if any of the following conditions are detected:

- The key of the base cluster does not match the copybook fields.
- The key of an alternate index does not match the copybook fields.
- The length of the copybook record is greater than the length of the base cluster record.

Assembler copybook notes

When the input copybook is in Assembler, the automated mapping facility assembles the copybook in foreground. If the return code from the assembly is not zero, the assembly analysis terminates, the assembly listing is written to a data set, and the dialog displays the listing.

If the copybook contains multiple DSECT statements, each structure is assumed to redefine the record.

Editing mapping information

Figure 41 on page 64 shows the mapping information for the CICS VT sample file VIDKSDS.

```
 --------------------- CICS VT: Edit columns in VIDKSDS  ------- Row 1 from 11
Command ===> ____________________________________________________ Scroll ===> CSR

Commands: MAP  SAVE  PREVIEW  SUSPEND/RESUME  CHANGE/UNDO  SHOW
Actions:  S Display, U Update, D Delete, I Insert            Status message /
A Pos   Copybook field name / DB2 column name     Type Len   Exit      Pic Par
- ----- ----+----1----+----2----+----3----+----   ---- ----- -----------------
_ 1     ITEM-NUMBER                             +      6
        ITEM_NUMBER...................          + CHAR 6
_ 7     ITEM-NAME                               +      12
        ITEM_NAME.....................          + CHAR 12
_ 19    ITEM-COLOUR                             +      6
        ITEM_COLOUR...................          + CHAR 6
_ 25    ITEM-HEIGHT                             +      4
        ITEM_HEIGHT...................          + CHAR 4
_ 29    ITEM-COST                               +      4
        ITEM_COST.....................          +  DEC 7,2
_ 33    ITEM-REORDER-QUANTITY                   +      3
        ITEM_REORDER_QUANTITY.........          +  DEC 3,0
_ 36    ITEM-SUPPLIER-CODE                      +      3
        ITEM_SUPPLIER_CODE............          + CHAR 3
_ 39    ITEM-DATE-FIRST-SHIP                    +      5
        ITEM_DATE_FIRST_SHIP..........          +  DEC 9,0
_ 44    ITEM-SHELF-LIFE                         +      2
        ITEM_SHELF_LIFE...............          + CHAR 2
_ 46    ITEM-DESCRIPTION                        +      35    Redefined field 1
        ITEM_DESCRIPTION..............          + CHAR 35
_ 46    ITEM-DETAILS-1                          +      35    Redefined field 1
        ITEM_DETAILS_1................          + CHAR 35
```

Figure 41. VIDKSDS Mapping

The copybook field name, position, and length are displayed. The DB2 column name, type, and length is shown beneath the field name. Overtype the column name if required.

There are a number of possible messages that CICS VT will show in the status field. The conditions that cause these messages to be displayed are as follows:

Table 13. Message display conditions

Status field value	**Copybook condition**
Name shortened	The length of a copybook field name exceeds 30 characters.
Name shortened	CICS VT adds a numeric suffix to the column name for each element in a field with an OCCURS clause. The length of the resulting column name exceeds 30 characters.
Duplicate name	The field name is duplicated within the copybook. This can occur when the same field name is specified in different group fields, or when a name shortened by CICS VT is duplicated.
Redefined field *n*	A field is either within a redefining or redefined area, where *n* is the redefinition level.
Field split	A field that is not the last field in the copybook exceeds 255 characters, which is the longest fixed length character column supported by DB2.

Table 13. Message display conditions (continued)

Status field value	Copybook condition
Initial numeric	A copybook field starts with a numeral and DB2 does not support column names starting with a numeral. The character # is added as the initial character of the column.
Unsigned binary	A COBOL copybook field attribute is unsigned binary. Field values in the VSAM file may exceed the maximum value supported by DB2 for the mapped column data type. See "Supported numeric formats" on page 88 for a discussion on numeric field types and the potential implications for unsigned binary fields.
Potential date	The COBOL copybook field has the conceptual data item DATE. You may want to update the mapping to use a DATE column data type in DB2.
Floating point	The COBOL copybook field has a floating-point picture. (COMP-1 or COMP-2). The auto-mapper maps these field types to CHAR(4)and CHAR(8) DB2 columns respectively. CICS VT does not support DB2 floating-point column data types.
DBCS	The COBOL copybook field has a DBCS picture. The auto-mapper maps this field to CHAR DB2 column. CICS VT does not support DB2 DBCS column data.
Assumed varying	The VIDMAPIN utility has detected that AVGLRECL is not equal to MAXLRECL in the base cluster. If the last copybook field is character data, the auto-mapper assumes that the file contains varying length records and specifies VARCHAR as the data type for the last column in the table. This status is also displayed if the last group or discrete field is an array with OCCURS DEPENDING ON.

You should note the following about the status field messages:

- You must analyze all the status field messages and perform an action where appropriate.
- The specific action or actions you need to take is based on the condition reported in the status field.
- CICS VT will not detect duplicate column names that occur as a result of you renaming a column.

There are 2 particular situations where manual intervention in the auto-mapping process is especially important:

1. You must understand how the automated mapping facility handles OCCURS and REDEFINES clauses before you decide what actions to perform. Refer to "COBOL REDEFINES clause and PL/I pointer based structures" on page 77 for more information about REDEFINES and OCCURS clauses.

2. There is no definitive characteristic or attribute to indicate that a VSAM file contains varying length records. Refer to "Variable length records in VSAM" on page 92 for more information about files containing varying length records.

Displaying a column

You display a column by specifying S on the command line. The following screen is displayed:

```
---------------------------- CICS VT: Display column ----------------------------
 Command ===> _____________________________________________ Scroll ===> CSR_

 Copybook field:
   name          : ZONED-DEC-01                                              +
   position      : 63
   length        : 8

 Column:
   name          : ZONED_DEC_01                                              +
   length        : 8,0
   type          : DEC

 Mapping:
   picture / FBE :
   parameters    :




 Press:  PF3 to exit   PF1 for help
```

Figure 42. Display column screen

Updating a column

Specify U on the command line to update a column. The following screen is displayed:

```
---------------------------- CICS VT: Update column ----------------------------
 Command ===> _____________________________________________ Scroll ===> CSR_

 Column selected: ITEM_DATE_FIRST_SHIP                                       +

 Copybook field length   ==> 5
 Current column length   ==> 9,0
 Current column type     ==> DEC


 Enter new column length ==> _____  (if DEC specify x,y)

 Enter new column type   ==> ____   (CHAR,DEC,INT,SINT,BINT,DATE,TIME,TIMS)

 Specify picture / FBE   ==> ________________ (example  HH.XX.SS.NNNNNN)
                                              (or MMDDYY)
                                              (or EXITx=exit name)

 Parameters              ==> ____________________________ Optional user parameters



 Press: Enter to confirm   PF3 to exit   PF1 for help
```

Figure 43. Update column screen

The column length and data type can be changed using the update column facility. The supported column types are displayed on the screen. If you specify a new data type of DATE, TIME, or TIMS, you are forced to define the appropriate picture string. This is shown in Figure 44:

```
---------------------------- CICS VT: Update column -----------------------------
 Command ===> _____________________________________________ Scroll ===> CSR_

 Column selected: ITEM_DATE_FIRST_SHIP                                        +

 Copybook field length   ==> 5
 Current column length   ==> 9,0
 Current column type     ==> DEC


 Enter new column length ==> _____  (if DEC specify x,y)

 Enter new column type   ==> DATE   (CHAR,DEC,INT,SINT,BINT,DATE,TIME,TIMS)

 Specify picture / FBE   ==> CCYYMMDD_________ (example  HH.XX.SS.NNNNNN)
                                                (or MMDDYY)
                                                (or EXITx=exit name)

 Parameters              ==> ______________________________ Optional user parameters



 Press: Enter to confirm   PF3 to exit   PF1 for help
```

Figure 44. Update column screen with sample values

See the *CICS VT Reengineering and Customization Guide* for information about defining picture strings.

CICS VT allows you to change the length of a column. You should carefully consider the potential effects of changing a column length to avoid adverse results. For example, if you specify a DB2 column length that exceeds the copybook field length, data retrieved from DB2 will potentially be truncated. This could be particularly significant if the data is numeric.

You can specify an FBE and add optional user parameters. These are described in "Mapping fields" on page 43 .

Deleting a column

Specifying `D` on the command line deletes a column, and produces the following confirmation screen:

```
                                  Confirm deletion


Command ===>                                            Scroll ====> CSR


    Column to be deleted: ZONED_DEC_01                                  +

    Press Enter to confirm delete
    Press PF3 to cancel delete
```

Figure 45. Delete column screen

Use the delete column option if you have filler fields or other fields that are obsolete and not required in DB2 .

Use the delete column option for redefined fields. The way that CICS VT handles redefined fields is explained in “COBOL REDEFINES clause and PL/I pointer based structures” on page 77 . Figure 46 shows how to use the delete column option to manage an example of redefined copybook fields:

```
 ---------------------- CICS VT: Edit columns in TFILE04 ------- Row 15 from 36
Command ===> ________________________________________________ Scroll ===> CSR

Commands: MAP  SAVE  PREVIEW  SUSPEND/RESUME  CHANGE/UNDO  SHOW
Actions:  S Display, U Update, D Delete, I Insert            Status message /
A Pos   Copybook field name / DB2 column name     Type Len   Exit      Pic Par
- ----- ----+----1----+----2----+----3----+----   ---- ----- -----------------
d 63    ZONED-DEC-01                          +      8     Redefined field 1
        ZONED_DEC_01.................         +  DEC 8,0
d 63    CHAR-REDEF-L1                         +      8     Redefined field 2
        CHAR_REDEF_L1................         + CHAR 8
_ 63    CHAR-REDEF-2X                         +      2     Redefined field 2
        CHAR_REDEF_2X................         + CHAR 2
d 65    CHAR-REDEF-2Y                         +      6     Redefined field 3
        CHAR_REDEF_2Y................         + CHAR 6
_ 65    CHAR-REDEF-3X                         +      2     Redefined field 3
        CHAR_REDEF_3X................         + CHAR 2
d 67    CHAR-REDEF-3Y                         +      4     Redefined field 4
        CHAR_REDEF_3Y................         + CHAR 4
_ 67    CHAR-REDEF-4X                         +      1     Redefined field 4
        CHAR_REDEF_4X................         + CHAR 1
d 68    CHAR-REDEF-4Y                         +      3     Redefined field 5
        CHAR_REDEF_4Y................         + CHAR 3
_ 68    CHAR-REDEF-5X                         +      1     Redefined field 5
        CHAR_REDEF_5X................         + CHAR 1
_ 69    CHAR-REDEF-5Y                         +      2     Redefined field 5
        CHAR_REDEF_5Y................         + CHAR 2
_ 71    DATE-DEC-02                           +      8     Redefined field 1
        DATE_DEC_02..................         +  DEC 8,0
_ 71    CHAR-REDEF-CC                         +      2     Redefined field 1
        CHAR_REDEF_CC................         + CHAR 2
```

Figure 46. Deleting redefined fields

After processing the actions, the 8 bytes starting at position 63 will be composed of the columns CHAR_REDEF_2X, CHAR_REDEF_3X, CHAR_REDEF_4X, CHAR_REDEF_5X, and CHAR_REDEF_5Y. This is shown in the Edit columns screen that follows:

```
 -------------------- CICS VT: Edit columns in TFILE04 ------- Row 15 from 31
Command ===> _____________________________________________ Scroll ===> CSR

Commands: MAP  SAVE  PREVIEW  SUSPEND/RESUME  CHANGE/UNDO  SHOW
Actions:  S Display, U Update, D Delete, I Insert            Status message /
A Pos   Copybook field name / DB2 column name     Type Len   Exit      Pic Par
- ----- ----+----1----+----2----+----3----+----   ---- ----- -----------------
_ 63    CHAR-REDEF-2X                           +      2     Redefined field 1
        CHAR_REDEF_2X.................          + CHAR 2
_ 65    CHAR-REDEF-3X                           +      2     Redefined field 2
        CHAR_REDEF_3X.................          + CHAR 2
_ 67    CHAR-REDEF-4X                           +      1     Redefined field 3
        CHAR_REDEF_4X.................          + CHAR 1
_ 68    CHAR-REDEF-5X                           +      1     Redefined field 4
        CHAR_REDEF_5X.................          + CHAR 1
_ 69    CHAR-REDEF-5Y                           +      2     Redefined field 4
        CHAR_REDEF_5Y.................          + CHAR 2
_ 71    DATE-DEC-02                             +      8     Redefined field 1
        DATE_DEC_02...................          +  DEC 8,0
_ 71    CHAR-REDEF-CC                           +      2     Redefined field 1
        CHAR_REDEF_CC.................          + CHAR 2
_ 73    CHAR-REDEF-YY                           +      2     Redefined field 1
        CHAR_REDEF_YY.................          + CHAR 2
_ 75    CHAR-REDEF-MM                           +      2     Redefined field 1
        CHAR_REDEF_MM.................          + CHAR 2
_ 77    CHAR-REDEF-DD                           +      2     Redefined field 1
        CHAR_REDEF_DD.................          + CHAR 2
_ 79    OCCUR-01                                +      10    Redefined field 1
        OCCUR_01......................          +  DEC 10,0
_ 79    OCCUR-FIELD-01_001                      +      2     Redefined field 1
        OCCUR_FIELD_01_001............          + CHAR 2
```

Figure 47. Edit columns screen after deleting redefined fields

You can specify delete for multiple columns at the same time. A delete confirmation panel is shown for each deleted column.

Inserting a new column

Consider the data set analyzed by the automated mapping facility in Figure 48 on page 70.

```
 --------------------- CICS VT: Edit columns in VIDKSDS2 ------- Row 4 from 14
Command ===> ____________________________________________________ Scroll ===> CSR

Commands: MAP  SAVE  PREVIEW  SUSPEND/RESUME  CHANGE/UNDO  SHOW
Actions:  S Display, U Update, D Delete, I Insert            Status message /
A Pos   Copybook field name / DB2 column name     Type Len   Exit      Pic Par
- ----- ----+----1----+----2----+----3----+----   ---- ----- -----------------
_ 25    ITEM-HEIGHT                            +      4
        ITEM_HEIGHT...................         + CHAR 4
_ 29    ITEM-COST                              +      4
        ITEM_COST.....................         +  DEC 7,2
_ 33    ITEM-REORDER-QUANTITY                  +      3
        ITEM_REORDER_QUANTITY.........         +  DEC 3,0
_ 36    ITEM-SUPPLIER-CODE                     +      3
        ITEM_SUPPLIER_CODE............         + CHAR 3
_ 39    FSHIP-CC                               +      2
        FSHIP_CC......................         +  DEC 2,0
_ 41    FSHIP-YY                               +      2
        FSHIP_YY......................         +  DEC 2,0
_ 43    FSHIP-MM                               +      2
        FSHIP_MM......................         +  DEC 2,0
_ 45    FSHIP-DD                               +      2
        FSHIP_DD......................         +  DEC 2,0
_ 47    ITEM-SHELF-LIFE                        +      2
        ITEM_SHELF_LIFE...............         + CHAR 2
_ 49    ITEM-DESCRIPTION                       +      35    Redefined field 1
        ITEM_DESCRIPTION..............         + CHAR 35
_ 49    ITEM-DETAILS-1                         +      35    Redefined field 1
        ITEM_DETAILS_1................         + CHAR 35
```

Figure 48. Inserting a new column screen

The fields FSHIP_CC, FSHIP_YY, FSHIP_MM and FSHIP_DD are each 2 bytes and represent a date. CICS VT lets you combine these fields and map them to a single DATE column in DB2. To do this requires the delete column and insert column functions.

You can only insert a new column or columns if the position in the record is not already assigned to a DB2 column. The four date fields are initially mapped to four 2-byte CHAR columns and they must be deleted before a new DATE column can be inserted. This is shown in Figure 49 on page 71.

```
 -------------------- CICS VT: Edit columns in VIDKSDS2 ------- Row 4 from 14
Command ===> _____________________________________________ Scroll ===> CSR

Commands: MAP  SAVE  PREVIEW  SUSPEND/RESUME  CHANGE/UNDO  SHOW
Actions:  S Display, U Update, D Delete, I Insert           Status message /
A Pos   Copybook field name / DB2 column name     Type Len   Exit      Pic Par
- ----- ----+----1----+----2----+----3----+----   ---- ----- -----------------
_ 25    ITEM-HEIGHT                             +      4
        ITEM_HEIGHT...................          + CHAR 4
_ 29    ITEM-COST                               +      4
        ITEM_COST.....................          +  DEC 7,2
_ 33    ITEM-REORDER-QUANTITY                   +      3
        ITEM_REORDER_QUANTITY.........          +  DEC 3,0
_ 36    ITEM-SUPPLIER-CODE                      +      3
        ITEM_SUPPLIER_CODE............          + CHAR 3
d 39    FSHIP-CC                                +      2
        FSHIP_CC......................          +  DEC 2,0
d 41    FSHIP-YY                                +      2
        FSHIP_YY......................          +  DEC 2,0
d 43    FSHIP-MM                                +      2
        FSHIP_MM......................          +  DEC 2,0
d 45    FSHIP-DD                                +      2
        FSHIP_DD......................          +  DEC 2,0
_ 47    ITEM-SHELF-LIFE                         +      2
        ITEM_SHELF_LIFE...............          + CHAR 2
_ 49    ITEM-DESCRIPTION                        +      35    Redefined field 1
        ITEM_DESCRIPTION..............          + CHAR 35
_ 49    ITEM-DETAILS-1                          +      35    Redefined field 1
        ITEM_DETAILS_1................          + CHAR 35
```

Figure 49. Deleting fields prior to inserting a new column

After the columns are deleted, you can insert the new DATE column. Specify I against the column name ITEM_SUPPLIER_CODE to enter the new column details. The details that you need to enter are shown in Figure 50:

```
 --------------------------- CICS VT: Insert column --------------------------
Command ===> _____________________________________________ Scroll ===> CSR

 Enter new field and column attributes:-

 Copybook field:
   position       ==> 39___
   length         ==> 8___
   data type      ==> C      (C,P,U,F,H,B)

 Column:
   name           ==> ITEM_DATE_FIRST_SHIPPED..................... +
   length         ==> _____    Optional (if DEC specify x,y)
   data type      ==> DATE     (CHAR,DEC,INT,SINT,BINT,DATE,TIME,TIMS)

 Mapping:
   picture / FBE  ==> CCYYMMDD_________  (example  HH.XX.SS.NNNNNN)
                                                  (or MMDDYY)
                                                  (or EXITx=exit name)
   parameters     ==> ____________________________  Optional user parameters

Press: Enter to confirm   PF3 to exit   PF1 for help
```

Figure 50. Insert new column details screen

The Insert column screen displays the position in the VSAM record that the new column will map to. It also displays the length of the area that is not mapped. Figure 50 shows the position is 39 and the length is 8. You must specify the data type of the VSAM field as one of the following values:

- C for character
- P for signed packed decimal
- U for unsigned packed decimal
- F for binary fullword, which is 4 bytes
- H for binary halfword, which is 2 bytes
- B for 2 binary fullwords, which is 8 bytes

Next you must specify details of the new column to be inserted. If the data type is DEC, you must specify a scale and precision in the **Column length** field. You do not have to specify a column length if the column data type is fixed length (INT, SINT, BINT, DATE, TIME, or TIMS).

You are prompted for a picture string if the **Column data type** is DATE, TIME, or TIMS. See "Specifying a picture string" on page 45.

Inserting multiple columns

About this task

Assume that you want to split the 35-byte field ITEM-DESCRIPTION shown in Figure 48 on page 70, into two columns of 4 bytes and 31 bytes. The procedure is as follows:

Procedure

1. Delete the fields ITEM-DESCRIPTION and its redefining field ITEM-DETAILS-1 using the action code `D`. This is explained in "Deleting a column" on page 67. After confirming the deletes, you are returned to the edit columns screen.
2. Insert the first new field using the action code `I`. CICS VT calculates the length of the field that is available and displays this as the length in the Insert column screen, shown in Figure 51.

```
 --------------------------- CICS VT: Insert column --------------------------
Command ===> ____________________________________________________ Scroll ===> CSR

 Enter new field and column attributes:-

 Copybook field:
   position       ==> 49___
   length         ==> 35__
   data type      ==> _      (C,P,U,F,H,B)

 Column:
   name           ==> ..................................................... +
   length         ==> _____    Optional (if DEC specify x,y)
   data type      ==> ____     (CHAR,DEC,INT,SINT,BINT,DATE,TIME,TIMS)

 Mapping:
   picture / FBE  ==> _________________  (example  HH.XX.SS.NNNNNN)
                                                 (or MMDDYY)
                                                 (or EXITx=exit name)
   parameters     ==> ______________________________  Optional user parameters

Press: Enter to confirm   PF3 to exit   PF1 for help
```

Figure 51. CICS VT: Insert column

3. Enter the new column name and attributes. Note that when you specify column type of CHAR, the **Column Length** field defaults to the value of the field length. In this example, you want to split ITEM-DESCRIPTION into two columns of 4 bytes and 31 bytes. The attributes of the first new column to be

inserted are shown in Figure 52:

```
 -------------------------- CICS VT: Insert column -------------------------
Command ===> ____________________________________________ Scroll ===> CSR

 Enter new field and column attributes:-

 Copybook field:
   position       ==> 49___
   length         ==> 4 __
   data type      ==> C      (C,P,U,F,H,B)

 Column:
   name           ==> ITEM_DESC_SHORT................................ +
   length         ==> _____    Optional (if DEC specify x,y)
   data type      ==> CHAR     (CHAR,DEC,INT,SINT,BINT,DATE,TIME,TIMS)

 Mapping:
   picture / FBE  ==> _________________  (example  HH.XX.SS.NNNNNN)
                                                  (or MMDDYY)
                                                  (or EXITx=exit name)
   parameters     ==> ____________________________  Optional user parameters

Press: Enter to confirm   PF3 to exit   PF1 for help
```

Figure 52. CICS VT: Insert column

4. Press Enter when you have supplied the column details. After the new column is inserted, you are returned to the Edit columns screen.
5. Insert the second column using the action code I. CICS VT calculates the field position and the available field length and displays them in the Insert column screen as shown in Figure 53.

```
 -------------------------- CICS VT: Insert column -------------------------
Command ===> ____________________________________________ Scroll ===> CSR

 Enter new field and column attributes:-

 Copybook field:
   position       ==> 53___
   length         ==> 31__
   data type      ==> _      (C,P,U,F,H,B)

 Column:
   name           ==> ............................................... +
   length         ==> _____    Optional (if DEC specify x,y)
   data type      ==> ____     (CHAR,DEC,INT,SINT,BINT,TIME,TIMS)

 Mapping:
   picture / FBE  ==> _________________  (example  HH.XX.SS.NNNNNN)
                                                  (or MMDDYY)
                                                  (or EXITx=exit name)
   parameters     ==> ____________________________  Optional user parameters

Press: Enter to confirm   PF3 to exit   PF1 for help
```

Figure 53. CICS VT: Insert column

6. Supply the attributes of the next new column and press Enter. For example, you could specify a column called ITEM_DESC_LONG that is 31 bytes in length, with column type CHAR. When the new column is inserted, you are returned to the Edit columns screen.
7. Confirm that the Edit columns screen contains your new columns. In the example, ITEM_DESC_SHORT and ITEM_DESC_LONG would display in the

Edit columns screen:

```
 --------------------- CICS VT: Edit columns in VIDKSDS2 ------- Row 1 from 11
Command ===> _____________________________________________ Scroll ===> CSR

Commands: MAP  SAVE  PREVIEW  SUSPEND/RESUME  CHANGE/UNDO  SHOW
Actions:  S Display, U Update, D Delete, I Insert            Status message /
A Pos   Copybook field name / DB2 column name     Type Len   Exit      Pic Par
- ----- ----+----1----+----2----+----3----+----   ---- ----- -----------------
_ 1     ITEM-NUMBER                           +      6
        ITEM_NUMBER...................        + CHAR 6
_ 7     ITEM-NAME                             +      12
        ITEM_NAME.....................        + CHAR 12
_ 19    ITEM-COLOUR                           +      6
        ITEM_COLOUR...................        + CHAR 6
_ 25    ITEM-HEIGHT                           +      4
        ITEM_HEIGHT...................        + CHAR 4
_ 29    ITEM-COST                             +      4
        ITEM_COST.....................        +  DEC 7,2
_ 33    ITEM-REORDER-QUANTITY                 +      3
        ITEM_REORDER_QUANTITY.........        +  DEC 3,0
_ 36    ITEM-SUPPLIER-CODE                    +      3
        ITEM_SUPPLIER_CODE............        + CHAR 3
_ 39    ITEM_DATE_FIRST_SHIPPED               +      8
        ITEM_DATE_FIRST_SHIPPED.......        + DATE                    Y
_ 47    ITEM-SHELF-LIFE                       +      2
        ITEM_SHELF_LIFE...............        + CHAR 2
_ 49    ITEM_DESC_SHORT                       +      4
        ITEM_DESC_SHORT...............        + CHAR 4
_ 53    ITEM_DESC_LONG                        +      31
        ITEM_DESC_LONG................        + CHAR 31
```

Automated mapping commands

The following commands are available during the automated mapping process: **MAP**, **SAVE**, **PREVIEW**, **SUSPEND**, **RESUME**, **SHOW**, and **CHANGE**.

MAP command

When you have completed field editing, you are ready to map the data set. You do this by specifying `MAP` on the command line. The mapping information is generated in the CICS VT system tables, and you are put into an ISPF edit of the generated DDL. You can manually update the DDL if required, you should *not* change the table name or column names, as this will cause the batch driver generation process to fail.

After you exit the ISPF edit session, a batch job is generated using the job card specified in Figure 38 on page 61, and automatically submitted for processing.

Before you issue the MAP command, make sure that you have taken appropriate actions for all columns with a status of `Redefined field`. Otherwise, the DB2 table will include columns for the redefined *and*the redefining fields.

SAVE command

Use the **SAVE** command when you have completed editing the fields. Like the **MAP** command, the **SAVE** command generates the mapping information in the CICS VT system tables, but the batch JCL to complete the mapping process is not submitted automatically. Instead the batch JCL is written to the data set *hlq*.VID.*dim_name*, where *hlq* is either your TSO prefix or the value you specified in “Specifying optional defaults panel 4” on page 36.

Use the **SAVE** command when you don't want to process the DDL immediately. For example, you might want to review some of the DDL parameters, or you might lack the necessary authority to create the objects in DB2. Manually submit the batch job at the appropriate time.

You should also use the **SAVE** command when you need to override the default DIM name for alternate indexes. See "DIM names for alternate indexes" on page 80.

PREVIEW command

The **PREVIEW** command lets you review the DDL that will be generated by the **MAP** and **SAVE** commands. Use this command at any time after the input copybook has been processed. For example if you delete redefined fields, use **PREVIEW** to see the DDL that will be generated.

The **PREVIEW** command can be abbreviated to **PV**.

SUSPEND command

The **SUSPEND** command saves all changes you have made for the current DIM, and ends the automated mapping process. All updates to the mapping are captured and stored in an ISPF table. Use **SUSPEND** in conjunction with the **RESUME** command described next.

SUSPEND is designed to support the mapping of one file at a time. It will not help you map multiple data sets in parallel. You can suspend mapping at any time by specifying SUSPEND on the command line.

RESUME command

The **RESUME** command restores the mapping information captured by the **SUSPEND** command using the ISPF table member VIDMCSUS. This allows you to continue mapping without losing updates. When you use the **RESUME** command, CICS VT does not check that the information captured by the **SUSPEND** command is for the data set that you are currently mapping. For example, if you map a data set called MYDIM and use the **SUSPEND** command to preserve your changes, then start mapping another data set called YOURDIM and use the **SUSPEND** command again, the changes you made to the mapping of MYDIM are lost.

The RESUME command is designed to support the mapping of one file at a time. It does not help you map multiple files in parallel.

SHOW command

Use the **SHOW** command to control the fields that are displayed in the Edit columns screen. To limit the display to those columns that have a message in the status area, specify either **SHOW ONLY** or **SHOW MESSAGES**. To redisplay all fields, use either **SHOW**, **SHOW ALL**, or **RESET**.

CHANGE command

The purpose of the **CHANGE** command is to perform a mass change of the generated column names. A REXX command shell is invoked and you provide one or more REXX statements to effect the desired change.

For example, assume that every copybook field name has a prefix of FITEM-. You want to strip the prefix from the generated column names. When you invoke the **CHANGE** command, the following screen is displayed:

```
---------------- CICS VT: Auto mapping mass change for VIDKSDS2 ----------------
 Command ===>                                                 Scroll ===> CSR

 Expression ===>




 DB2 column     ===> FITEM_FSHIP_DD                                          +
 Copybook field ===> FITEM_FSHIP-DD                                          +
                     --------1--------2--------3
 New DB2 column ===> FITEM_FSHIP_DD                                          +

 Write an expression using REXX which sets CNAME. CICS VT provides variables:
   CNAME DB2 column name     CNAMEL length of DB2 column name
   FNAME copybook field name FNAMEL length of copybook field name
 CICS VT will automatically translate "-" to "_" in your copybook field names.

 For example, to remove a 6-byte common prefix: CNAME = SUBSTR(FNAME,7)
              to add a prefix of "VT_"          CNAME = 'VT_' || CNAME

 Press: Enter to preview your edit  or  PF3 to process your edit
 CANCEL will cancel the edit, leaving the field names unchanged
```

Figure 54. Change command input screen

In the Expression area, the following REXX command strips the FITEM_ prefix from all of the generated column names:

```
CNAME = SUBSTR(FNAME,7)
```

Type **CANCEL** to undo the changes and return to the Edit columns screen.

You can write a short REXX program for more complex changes. For example, the following program removes the string XXXX- from anywhere within the field name:

```
S = 'XXXX-'; L = LENGTH(S); CNAME = FNAME; X = POS(S,CNAME);
DO WHILE X > 0;
   IF X = 1 THEN CNAME = SUBSTR(CNAME,L+1);
   ELSE CNAME = LEFT(CNAME,X-1) || SUBSTR(CNAME,X+L);
   X = POS(S,CNAME);        END;
```

You can specify a maximum of 383 characters of REXX.

Note the following factors when you are using the **CHANGE** command:

- The REXX is applied against all fields and columns unless you include logic to limit changes. Avoid REXX that makes unconditional changes.
- FILLER fields seldom have a prefix. Write your REXX to avoid changing FILLER fields.
- Apart from the **CANCEL** command, there is currently no facility to undo changes. Only press PF3 when you have reviewed and accepted the effects of the **CHANGE** command against all columns.

For very complex transformations, you can write your own logic in a REXX library, and call it from the **CHANGE** command. Here is an example of how to call the REXX function MYREXX:

```
CNAME = MYREXX(FNAME,'string')
```

There is a sample REXX program called VIDEDIT for use with the **CHANGE** command in *cvt210.*.SVIDEXEC.

COBOL OCCURS clause and PL/I arrays

CICS VT generates multiple columns for OCCURS clauses and arrays. For example, a copybook field defined as PAY-MONTH OCCURS 12 would become 12 columns in DB2 called PAY_MONTH_001, PAY_MONTH_002, and so on up to PAY_MONTH_012.

If the OCCURS field name is longer than 26 characters, the column names generated by CICS VT exceed the DB2 30-byte limit when the 3-byte sequence number is added. In this case the status for the *first* field will be set to `Name shortened`, but the status for the remaining fields in the group will be blank.

You should be aware of the implications of variable length VSAM records and how they potentially affect your DB2 table design. This topic is covered in "Variable length records in VSAM" on page 92.

COBOL REDEFINES clause and PL/I pointer based structures

CICS VT generates a DB2 column for every copybook field. If a field is redefined, a column is generated for the original field and the redefining field. To illustrate this, look at the following extract of a COBOL copybook:

```
05 ZONED-DEC-01               PIC 9(8)              .
05 CHAR-REDEF-L1 REDEFINES  ZONED-DEC-01 PIC X(8)   .
05 CHAR-REDEF-L2 REDEFINES    CHAR-REDEF-L1         .
   07 CHAR-REDEF-2X           PIC X(02)             .
   07 CHAR-REDEF-2Y           PIC X(06)             .
   07 CHAR-REDEF-L3 REDEFINES CHAR-REDEF-2Y         .
       09 CHAR-REDEF-3X       PIC X(02)             .
       09 CHAR-REDEF-3Y       PIC X(04)             .
       09 CHAR-REDEF-L4 REDEFINES CHAR-REDEF-3Y     .
          11 CHAR-REDEF-4X    PIC X(01)             .
          11 CHAR-REDEF-4Y    PIC X(03)             .
          11 CHAR-REDEF-L5 REDEFINES CHAR-REDEF-4Y  .
             13 CHAR-REDEF-5X PIC X(01)             .
             13 CHAR-REDEF-5Y PIC X(02)             .
```

The edit columns screen that CICS VT produces for this copybook is shown in Figure 55 on page 78:

```
 ---------------------- CICS VT: Edit columns in TFILE04 ------- Row 15 from 36
Command ===> _____________________________________________ Scroll ===> CSR

Commands: MAP  SAVE  PREVIEW  SUSPEND/RESUME  CHANGE/UNDO  SHOW
Actions:  S Display, U Update, D Delete, I Insert            Status message /
A Pos   Copybook field name / DB2 column name     Type Len   Exit      Pic Par
- ----- ----+----1----+----2----+----3----+----   ---- ----- -----------------
_ 63    ZONED-DEC-01                          +      8     Redefined field 1
        ZONED_DEC_01.................         +  DEC 8,0
_ 63    CHAR-REDEF-L1                         +      8     Redefined field 2
        CHAR_REDEF_L1................         + CHAR 8
_ 63    CHAR-REDEF-2X                         +      2     Redefined field 2
        CHAR_REDEF_2X................         + CHAR 2
_ 65    CHAR-REDEF-2Y                         +      6     Redefined field 3
        CHAR_REDEF_2Y................         + CHAR 6
_ 65    CHAR-REDEF-3X                         +      2     Redefined field 3
        CHAR_REDEF_3X................         + CHAR 2
_ 67    CHAR-REDEF-3Y                         +      4     Redefined field 4
        CHAR_REDEF_3Y................         + CHAR 4
_ 67    CHAR-REDEF-4X                         +      1     Redefined field 4
        CHAR_REDEF_4X................         + CHAR 1
_ 68    CHAR-REDEF-4Y                         +      3     Redefined field 5
        CHAR_REDEF_4Y................         + CHAR 3
_ 68    CHAR-REDEF-5X                         +      1     Redefined field 5
        CHAR_REDEF_5X................         + CHAR 1
_ 69    CHAR-REDEF-5Y                         +      2     Redefined field 5
        CHAR_REDEF_5Y................         + CHAR 2
_ 71    DATE-DEC-02                           +      8     Redefined field 1
        DATE_DEC_02..................         +  DEC 8,0
_ 71    CHAR-REDEF-CC                         +      2     Redefined field 1
        CHAR_REDEF_CC................         + CHAR 2
```

Figure 55. Edit column screen for redefined fields

The screen shows that the column ZONED_DEC_01 which corresponds to the copybook field ZONED-DEC-01 is at position 63 in the VSAM record, has a length of 8 bytes, and has a status of `Redefined field` 1. The number to the right of the status field indicates the level number of the redefined field.

The column CHAR_REDEF_L1 also starts at position 63 and has a length of 8 bytes, but the level number is 2. This is because it is also redefined, and the redefining fields correspond to columns CHAR_REDEF_2X and CHAR_REDEF_2Y.

The starting position and the redefined field level number are the key factors to be assessed to help you decide what action you should perform. All of the fields in Figure 31 are part of the 8 bytes of the VSAM record starting at position 63. You must decide which column or columns will contain the data for these 8 bytes, and use the delete column option to remove the remaining fields.

CICS VT supports a maximum of five levels of redefined fields within a group item.

Automated mapping batch jobs

The batch job that maps the data set submits one or more additional batch jobs. The number and content of these jobs depends on the characteristics of the data set being processed. There are a minimum of two batch jobs, and a further job for each alternate index path associated with the base cluster. These jobs are explained in more detail next.

Automated mapping job 1

The first job performs the following functions:

Step number	Step name	Function
1	PATHMAP	Generates alternate index mapping information. This step is only processed if one or more alternate indexes exist.
2	DDLSUB	Executes the DB2 sample program DSNTEP2 to process the DDL for all of the DB2 tablespace, table, and index objects.
3	GEN	Generates the DIM for the base cluster.
4	ASM	Assembles the base cluster DIM.
5	LKED	Link-edits the base cluster DIM.
6	CLEANUP	Deletes temporary data sets.
7	VIDDDMG	Generates the data for the DDM for the base cluster, and for each alternate index.
8	SUBMIT	Executes program IKJEFT01, which submits the jobs to produce the DDMs for the base cluster and alternate indexes.

Steps 3, 4, 5, and 6 are repeated instream for each alternate index path on the base cluster. Step 8 submits one job for the base cluster DDM, and one job for each alternate index DDM. All of the steps should produce a return code of zero.

Ensure that the TSO user ID performing the automated mapping has the appropriate authorities and privileges to create all of the DB2 objects.

Automated mapping job 2

The batch jobs to create the DDMs have the same job name and are therefore serialized. The job steps are as follows:

Step number	Step name	Function
1	SQ000000	Extracts the DDM mapping data produced by step 7 in automated mapping job 1.
2	PRECOMP	Pre-compiles the assembler source using the DB2 DSNHPC program and creates the DB2 DBRM.
3	PRECOMP	Splits the assembler produced by step 2 into multiple CSECTS.
4	ASMB001	Assembles the CSECT for the runtime control blocks.
5	ASMB002	Assembles the CSECT for SQL forward processing.
6	ASMB003	Assembles the CSECT for SQL backward processing.
7	LKED	Link-edits the CSECTs from the preceding assembly steps.
8	BIND	Binds the DBRM from step 2 into a DB2 package, and binds the application plan for the DDM.

The number of assembly steps is a factor of the value you specify for the number of concurrent threads parameter, which is explained in “Specifying mandatory defaults panel 1” on page 32. If you specify a value of 5, there will be eleven assembly steps.

Step 2 always ends with a return code of 04. Step 7 also might end with a return code of 04, but this is dependent on z/OS system defaults. All of the other steps should produce return code 00.

DIM names for alternate indexes

The automated mapping facility generates DIMs and DDMs for each alternate index path. The module names use the first four characters of the base cluster DIM name, followed by X01 for the first alternate index DIM, X02 for the second and so on.

CICS VT allocates DIM names for alternate index paths in the same sequence that the paths were defined to VSAM. For example, consider the following extract from the list catalog command for the base cluster VID.VIDKSDS:

```
ASSOCIATIONS
   DATA-----VID.VIDKSDS.DATA
   INDEX----VID.VIDKSDS.DATA.INDEX
   AIX------VID.VIDKSDS.DATA.ZIP
   AIX------VID.VIDKSDS.DATA.NAME
   AIX------VID.VIDKSDS.DATA.ADDRESS
```

There are three paths defined for this data set. The list catalog command shows these in the order that they were defined. If the DIM name for the base cluster is VIDKSDS, the DIM names generated for the alternate index paths are as follows:

Name	Function
VIDKX01	The DIM for the ZIP path.
VIDKX02	The DIM for the NAME path.
VIDKX03	The DIM for the ADDRESS path.

If two base cluster DIMs have the same first four characters and have alternate indexes, SQL code -803 occurs in the PATHMAP step. In this case, choose an alternative DIM name.

There are two methods for choosing an alternate index DIM name:

1. The first method uses the VIDMAIX REXX exit. You develop this exit to override the default DIM name generated by automated mapping for alternate indexes.
2. The second method requires the following actions:
   a. Use the SAVE command, which preserves a copy of the mapping JCL in a data set that you can modify.
   b. Modify the PATHNAME parameter in the generated data set *prefix*.VID.ISPAIX and specify the new AIX DIM name.
   c. Modify the generated data set *prefix*.VID.*dim-name* , where *dim-name* is the base cluster DIM name, and change all occurrences of the default AIX DIM name to the new DIM name.

Read “Automated mapping commands” on page 74 for information on the **SAVE** command.

Post auto-mapping tasks

Before proceeding to the data migration activities, there are two optional tasks that you can perform, and these are described next.

Capture generated DDL

To be able to process a migrated data set on a different DB2 system, you must create the DB2 objects that were created in step 2 of the automated mapping job 1. To do this, you should capture the DDL that was used as input to this step and store it in a separate data set or library. You specify the data set name containing the DDL in the auto mapping input panel as shown in Figure 40 on page 62. You might have to tailor parts of the DDL, such as the database and storage group names, to conform to your new DB2 system requirements.

Verify DDM access paths

Verify that the DB2 access paths for the DDMs are correct. To generate the PLAN_TABLE data in DB2 , you specify `EXPLAIN YES` in the BIND model in SVIDSAMP field as shown in Figure 13 on page 33 . Alternatively, you can rebind the DDM package and add the parameter EXPLAIN YES. The PLAN_TABLE rows should be as follows:

- All DELETE statements should have an ACCESSTYPE of R.
- All INSERT statements should have an unspecified ACCESSTYPE.
- All other rows should have an ACCESSTYPE of I.
- There should be no sort activity.

You might have to refer to the output from the PRECOMP step in automated mapping job 2 to get the statement numbers referred to in the PLAN_TABLE.

Data migration

CICS VT includes utility programs to perform the initial conversion of your data from VSAM to DB2. The data migration process has four steps. The JCL and control cards for each step can be automatically generated using option *3* from the CICSVT ISPF main menu screen. This process is described in "Automated data migration."

Alternatively, you can manually create the migration jobs following the instructions in "Creating data migration jobs manually" on page 85.

Automated data migration

You can automatically create the appropriate data migration JCL and control cards using the CICS VT data migration generating utility. The utility is invoked from the ISPF main menu option 3.

```
 ----------------------------- CICS VT: Main menu ----------------------------

Select Option ===> 3_____________________________________________________

      1 - Auto mapping facility

      2 - Manual mapping facility

      3 - Generate migration jobs for DIM : VIDKSDS_ (Full name only)

      D - Defaults









Press: Enter to continue   PF3 to exit   PF1 for Help
```

Figure 56. CICS VT main menu

You must specify the full DIM name for option 3.

Specify output data sets

Option 3 from the ISPF main menu verifies that the DIM is mapped and presents the screen shown in Figure 57.

```
 ----------------------- CICS VT create migration jobs -----------------------
Command ===> ____________________________________________________________

DIM name                    : VIDKSDS

VSAM data set cluster       : appl.vsam.cluster_____________________________

Migration JCL library       : appl.VIDKSDS.migrate__________________________

Migration data sets prefix  : appl.VIDKSDS.temp_____

Migration DB2 load cards    : appl.VIDKSDS.syspunch_________________________

SUBMIT or EDIT JCL          : S____

Job card statement:
 //mig210 JOB VT,CLASS=a,MSGCLASS=a,NOTIFY=mytsoid_____________________
 //WHERESMY JCLLIB ORDER=cvt210.SVIDCNFG.cust___________________
 ____________________________________________________________________
 ____________________________________________________________________

 Press Enter to continue, PF3 to Exit or PF1 for Help                    Help
```

Figure 57. CICS VT create migration jobs

Supply the names of the output data sets that the automated data migration facility generates. The field descriptions are as follows:

Table 14. Field descriptions

Field	Description
VSAM data set cluster	If the file to be migrated was manually mapped, you must provide the name of the VSAM data set base cluster. If you mapped the DIM using automated mapping, this field displays the name of the data set you specified in the first step of the mapping. You can change the data set name if required.
Migration JCL library	This is a new output library created by the data migration utility containing various JCL and control information members. Specify the full data set name without quotes. Consider including the DIM in the data set name.
Migration data sets prefix	This defines the prefix for the transient data sets used during the data migration process. The maximum length of this parameter is 26 characters. Consider including the DIM in the data set name.
Migration DB2 load cards	This new sequential data set contains the input statement to the DB2 load utility. Consider including the DIM in the data set name.
SUBMIT or EDIT JCL	Specify either `S(ubmit)` or `E(dit)`. If you specify `EDIT`, you must submit the generated job manually.

Generation utility

The data migration batch job JCL and control cards are generated by the data migration generation utility.

If you specify the `E(dit)` option, you can review the generated JCL prior to submission. Here is an example of the generated JCL:

```
//*-------------------------------------------------------------------
//* GENERATE DIM MIGRATION JOBS FOR CICS VT
//* (C) COPYRIGHT 2011
//*     CIRCLE COMPUTER GROUP LIMITED
//*     ALL RIGHTS RESERVED
//*-------------------------------------------------------------------
//ACCTFIL  EXEC PGM=VIDMIGGN,PARM='db2ssid,VIDKSDS'
//STEPLIB  DD  DSN=appl.driver.load,
//         DISP=SHR
//         DD  DSN=VID.SVIDLODE,
//         DISP=SHR
//REPORT   DD  SYSOUT=*
//MIGOUT   DD  DSN=appl.VIDKSDS.migrate,
//         DISP=(MOD,CATLG,CATLG),
//         UNIT=SYSDA,SPACE=(TRK,(1,1)),
//         LRECL=80,BLKSIZE=0,DSNTYPE=LIBRARY,
//         DSORG=PO
//MIGPUNCH DD  DSN=appl.VIDKSDS.syspunch,
//         DISP=(MOD,CATLG,CATLG),
//         UNIT=SYSDA,SPACE=(TRK,(1,1)),
//         LRECL=135,BLKSIZE=0
//MIGPARM  DD  *
```

```
J1//mig210 JOB VT,CLASS=a,MSGCLASS=a,NOTIFY=mytsoid
J2//WHERESMY JCLLIB ORDER=cvt210.SVIDCNFG.custom
VSappl.vsam.cluster
PXappl.VIDKSDS.temp
```

You can change the generated JCL if required but avoid changing the control cards that follow the MIGPARM DD statement.

Theoretically, you could use the ISPF option to create the generation utility the first time only. Thereafter, you could edit the JCL and change all occurrences of the initial DIM to a new DIM and submit it manually.

Generated migration members

The migration library *appl*.VIDKSDS.*migrate* contains the following members:

VID1UNLD
: Batch job that executes the VIDUNLOD utility to unload a VSAM data set to a sequential file.

VID2LOAD
: Batch job that executes the VIDLOAD utility to read the unloaded VSAM data and create an output sequential data set formatted for DB2.

VID3DB2L
: Batch job that executes the DB2 load utility to load the VID2LOAD output data set, using the generated load statement in *appl*.VIDKSDS.*syspunch*.

VID4DUMM
: Batch job that executes the access method services IDCAMS utility to create the dummy VSAM data set required for CICS VT batch processing.

VID5DMF
: Batch job that executes the VIDREAD utility and uses DMF to verify successful mapping and data migration.

VID9CSD
: Input to the CICS CSD batch update utility to define the base cluster DIM and DDM, plus mapped FBEs and IRDs.

VID9DST
: The macro statements to add the base cluster DIM to the CICS VT DST.

The sequential data set *appl*.VIDKSDS.*syspunch* is the input control statement for the IBM DB2 load utility. Review the generated statement prior to running the load utility.

Migration utility notes

Some of the parameters used by the batch jobs generated by the data migration utility are based on certain assumptions. The following information should be reviewed prior to executing the generated jobs:

- The default CICS VT subsystem VIDS is used in VID5DMF. If you are not using the default subsystem name, you must update this member.
- The DB2 subsystem specified in the VIDMIGGN parameter list refers to the subsystem for the mapping tables.
- The DB2 subsystem specified in members VID3DB2L and VID5DMF refers to the subsystem for the migrated VSAM data.
- You must manually update the FILE= parameter in member VID9CSD, and specify the name of the file that is defined in CICS file control.

- The migration utility loads the VIDDDMEX module to determine the name of the DDM, which is used in members VID2LOAD, VID5DMF, and VID9CSD. If you are using your own modified version of VIDDDMEX, ensure that the order of STEPLIB/JOBLIB data sets means your modified version is used.
- If you are migrating a manually mapped VSAM data set, you have to update the space parameters in members VID1UNLD and VID2LOAD.
- The DD statements for the DB2 load utility in VID3DB2L do not consider the need to restart the utility in the event of a failure. If you are migrating a large VSAM file, consider modifying the JCL to enable restart.
- The members VID2LOAD and VID5DMF use the CICS VT drivers library, and the data set name is from the initial customization. If this is not the correct library, update the JCL procedures VIDLODP and VIDDMFP in *my*.SVIDCNFG.*custom*.
- CICS VT support for DB2 nullable columns requires the null value to be the first byte of the data field. Do not modify the `NULLIF` clause if present in the generated load card statement.

Creating data migration jobs manually

Although the recommended approach is to use the data migration generation utility, you can manually create the appropriate JCL and control cards for each of the four migration steps. The following sections describe each step and provide appropriate examples.

Step 1 - Unload the VSAM file

Use the CICS VT utility program VIDUNLOD to unload the VSAM file to a sequential data set.

Sample JCL for this utility can be found in member VIDUNLOD in VID.SVIDSAMP, as follows:

```
//VIDUNLOD EXEC PGM=VIDUNLOD
//STEPLIB   DD DSN=VID.SVIDLODE,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//FILEIN    DD DSN=appl.base.cluster.file,DISP=SHR
//FILEOUT   DD DSN=appl.sequential.unload,UNIT=SYSDA,
//             DISP=(,CATLG,DELETE),RECFM=FB,LRECL=108,
//             SPACE=(CYL,(25,5),RLSE)
```

Figure 58. Sample JCL for VIDUNLOD

Note the following points about the VIDUNLOD utility:

- The FILEOUT output file produced by VIDUNLOD must be fixed block.
- CICS VT prefixes each data record with 8 bytes of control information, containing the VSAM data set type and the length of each unloaded record. The record length of FILEOUT must be the maximum VSAM record length plus 8 bytes. Use the MAXLRECL parameter value from the IDCAMS LISTCAT utility program to determine the maximum record length of a VSAM data set.
- The VIDUNLOD utility can only be run against the base cluster data set.

When the VSAM file has finished unloading, the following message is sent to the JES job log:

```
+VIDUNLOD
+VIDUNLOD - HAS SUCCESSFULLY UNLOADED FILEIN TO FILEOUT
+VIDUNLOD
```

Step 2 - Convert the unloaded data

Use the CICS VT batch utility program VIDLOAD to convert the unloaded data set produced by the VIDUNLOD utility program into the correct format for the DB2 load utility.

Sample JCL for this utility can be found in member VIDLOAD in VID.SVIDSAMP, as follows:

```
//VIDLOAD   EXEC PGM=VIDLOAD,PARM=dimname
//STEPLIB   DD DSN=appl.drivers.load,DISP=SHR
//          DD DSN=VID.SVIDLODE,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//LOADIN    DD DSN=appl.sequential.unload,DISP=SHR
//LOADOUT   DD DSN=appl.sequential.db2load,UNIT=SYSDA,
//             DISP=(,CATLG,DELETE),RECFM=FB,LRECL=100,
//             SPACE=(CYL,(25,5),RLSE)
```

Figure 59. Sample JCL for VIDLOAD

Note the following points about the VIDLOAD utility:

- The name of the DIM is a required parameter for VIDLOAD. This is shown as *dimname* in the sample JCL.
- The output record format must be fixed block.
- The record length of the output file is the length of the DB2 row. If you are using exits to split a single data set into multiple DB2 tables, see "Conversion utility for mapping to multiple DB2 tables" on page 95.

The record length for the DB2 load file must consider any data re-engineering that affects the length of the DB2 row. For example, assume your VSAM record length is 100 bytes and contains a 6 byte unsigned zoned decimal field mapped to a DEC(6,0) column. DEC(6,0) is 4 bytes, so the DB2 row is 2 bytes less than the VSAM record. If, however, your file has a 5-byte packed decimal date field mapped to a DATE column, which is 10 bytes in DB2, the row is 5 bytes longer than the VSAM record. The most reliable method to establish the length of the DB2 row is to review the load utility input statement, and add the length of the last column to its position. In Figure 60 on page 87, the last column ITEM_DESCRIPTION is 35 bytes starting in position 50. The DB2 row length in this case is 84 bytes.

```
LOAD DATA LOG NO INDDN SYSREC00 NOCOPYPEND
    INTO TABLE
    CIRDL.VID_ITEM
 (
 ITEM_NUMBER                          POSITION(       1         )
 CHAR(                    6) ,
 ITEM_NAME                            POSITION(       7         )
 CHAR(                   12) ,
 ITEM_COLOUR                          POSITION(      19         )
 CHAR(                    6) ,
 ITEM_WEIGHT                          POSITION(      25         )
 CHAR(                    4) ,
 ITEM_COST                            POSITION(      29:      32)
 DECIMAL                     ,
 ITEM_REORDER_NO                      POSITION(      33         )
 SMALLINT                    ,
 ITEM_SUPP_CODE                       POSITION(      35         )
 CHAR(                    3) ,
 ITEM_DATE_FSHIP                      POSITION(      38         )
 DATE EXTERNAL(          10) ,
 ITEM_SHELF_LIFE                      POSITION(      48         )
 CHAR(                    2) ,
 ITEM_DESCRIPTION                     POSITION(      50         )
 CHAR(                   35) )
```

Figure 60. DB2 load statement

The VIDLOAD utility issues the following message at the end of the job:

```
+VIDLOAD
+VIDLOAD - DB2 LOADFILE FOR VSAM WRITTEN TO LOADOUT
+VIDLOAD
```

Step 3 - Load the data in DB2

This step uses the standard DB2 LOAD utility to load the output data set from the VIDLOAD program into DB2 .

The member VIDDB2LD in *my*.SVIDSAMP.*custom* contains sample JCL for the DB2 LOAD utility. You can use the DB2 DSNTIAUL sample program, or an equivalent program, to generate the DB2 load control cards. The *my*.SVIDSAMP.*custom* library member VIDTIAUL contains sample JCL for the DSNTIAUL utility.

Step 4 - Verify data migration

You can verify that the data has been migrated successfully using the dual-mode facility (DMF), which is described in Chapter 6, “The dual mode facility (DMF),” on page 111, and the VIDREAD utility. Sample JCL using the VIDREAD utility program follows:

```
//CHECKIT EXEC PGM=VIDREAD
//STEPLIB   DD DSN=appl.drivers.load,DISP=SHR
//          DD DSN=VID.SVIDLODE,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//FILEINV   DD DSN=appl.base.cluster.file,DISP=SHR
//FILEIN@   DD DSN=appl.base.cluster.file.DUMMY,DISP=SHR
//FILEIN    DD SUBSYS=(ssi,db2id,dimname,FILEINV)
//VIDTRCE   DD SYSOUT=*
//VIDTRCEP  DD *
TRACE P01,P03,P04,P05,P08,DIM=dimname
//
```

Figure 61. Using VIDREAD with DMF to verify data migration

The DD statement FILEINV defines the original VSAM data set and is specified as the fourth SUBSYS parameter to enable DMF. You can choose any DD name. Note

that you must use different VSAM data sets for FILEIN@ and FILEINV to avoid potential VSAM sharing issues. See “Additional DD statement for dummy VSAM data set” on page 98 for details concerning the FILEIN@ DD statement.

Additional data migration tasks

After you have loaded the data to DB2, you must perform a number of routine DB2 database administration tasks.

Some or all of these will be dictated by your current DB2 DBA standards.

1. Run the DB2 COPY utility to create an image copy and remove the copy pending flag. This is not required if your DB2 load utility includes NOCOPYPEND.
2. Check to see if the DB2 LOAD has caused multiple extents in either the tablespace or indexspace page sets. If necessary, use the ALTER SQL statement to modify the page set SPACE allocation.
3. To resolve multiple data set extents, run the DB2 REORG utility and gather RUNSTATS data. Otherwise, run RUNSTATS.
4. Optionally, rebind the DDM package with `EXPLAIN(YES)`.
5. Verify that the correct access paths have been selected by DB2.

Supported numeric formats

There are various combinations of signed and unsigned numeric fields that are valid in z/OS. VSAM does not impose any rules governing the format of numeric fields, but rules are enforced by DB2. Additionally, DB2 stores all DECIMAL, SMALLINT, INTEGER, and BIGINT column values as signed. This has potential implications for CICS VT. This section defines the combination of VSAM field type and DB2 column data type directly supported by CICS VT.

Zoned decimal fields

A zoned decimal field can be signed or unsigned. The valid sign representations for zoned decimal are as follows:

Positive
C, A, E, and F.

Negative
D and B.

Here are examples of signed and unsigned field attributes in COBOL showing the typical representation of the numeric value +123:

- PIC 9(3) is unsigned zoned decimal and the value is X'F1F2F3'
- PIC S9(3) is signed zoned decimal and the value is X'F1F2C3'

When you map a zoned decimal field to any numeric column in CICS VT, you specify the field type `C`. For every mapped combination of field type `C` and a DB2 numeric column, CICS VT returns the data to your application programs as unsigned. This applies to all DB2 numeric columns supported by CICS VT, which is currently DECIMAL,SMALLINT, INTEGER, and BIGINT.

If your VSAM file has a signed zoned decimal field, your program logic may depend on signed zoned decimal values. For example, a COBOL test for NUMERIC for a signed zoned decimal field is NOT TRUE if the field value is

unsigned. CICS VT has no way of knowing if your application code is dependent on signed zoned decimal values. Your options are to code a simple FBE to reset the field value to signed format, or change the DB2 column data type to CHARACTER. No data conversion is performed for a type C VSAM field that is mapped to a CHAR column.

Packed decimal fields

Packed decimal fields can be signed or unsigned. The valid sign representations are the same as for zoned decimal field. When you map a packed decimal field in in CICS VT, you specify a field type of P for *signed* packed decimal or U for *unsigned* packed decimal.

An all cases when a signed packed decimal field is mapped to a DB2 numeric column, the sign representation that CICS VT returns to your application is as follows:

- Signed positive is B'1100' ('C')
- Signed negative is B'1101' ('D')

An all cases when an unsigned packed decimal field is mapped to a DB2 numeric column, the sign representation that CICS VT returns to your application is as follows:

- Unsigned using B'1111' ('F')

CICS VT uses the same sign representation rules for packed decimal fields that map to any supported DB2 numeric column type.

Although the same application logic dependency issues for zoned decimal fields apply to packed decimal fields, you are only likely to encounter an issue if the same field has signed values in some records and unsigned values in other records. If you decide to change the column type to CHAR in this situation, you must also update the CICS VT field type to C in manual mapping. Otherwise the DIM driver generation will fail.

Binary fields

Binary fields can be signed or unsigned. In COBOL, binary fields are defined as COMP, COMP-4, or COMP-5. CICS VT supports binary fields mapped to DB2 column types of SMALLINT, INTEGER, and BIGINT. DB2 data in column types of SMALLINT, INTEGER, and BIGINT is always signed.

CICS VT field types H, F, and B correspond to binary field lengths of 2 bytes, 4 bytes, and 8 bytes respectively. The range of data values in signed binary fields is consistent with the values in the DB2 binary data types SMALLINT, INTEGER, and BIGINT. A potential issue exists with unsigned binary fields. For example, the maximum value in a signed halfword binary field is +32767. This is the same for a DB2 SMALLINT column. The maximum value in an unsigned halfword binary field is 65535. In some cases, it may be necessary to map field type H to an INTEGER column, or a field type F to a BIGINT column.

The automated mapping facility highlights instances of unsigned binary copybook fields. You must know the range of values in unsigned binary fields in your VSAM files to ensure that you map them to appropriate column data types.

You can avoid potential issues with binary field data values by mapping them to CHAR columns. If so, you must update the CICS VT field type to `C` in each case.

Floating point

CICS VT does not support the mapping of floating point data in VSAM to the single or double precision floating point data types in DB2. Floating point fields must be mapped to CHAR columns of the same length, and must have the `FOR BIT DATA` attribute.

The auto-mapper recognises floating point fields defined in COBOL copybooks and generates appropriate mapping and DDL.

Partial keys for POINT, START, and STARTBR

VSAM field values that are not consistent with the mapped DB2 column type will come to light during the data migration process. However, you may also encounter runtime data issues with programs that use a partial key value or a generic key in a positioning call. For example, assume that you have VSAM file with a group field key as follows:

```
01  SAMPLE-REC.
    05  SAMPLE-KEY.
        10  SAMPLE-KEY-CHAR      PIC X(12).
        10  SAMPLE-KEY-NUMB      PIC 9(4).
    05  SAMPLE-DATA              PIC X(1000).
```

The field SAMPLE-KEY-CHAR is defined as type `C` and is mapped to a CHAR(12) column. The field SAMPLE-KEY-NUMB is defined as type `C` and is mapped to a DEC(4,0) column. If a COBOL program performs MOVE SPACES TO SAMPLE-KEY and then MOVE *value* TO SAMPLE-KEY-CHAR, an EXEC CICS STARTBR gets a data exception error because CICS VT cannot convert a value of spaces in SAMPLE-KEY-NUMB to a valid numeric. The solution in this situation is to code a simple FBE that tests for spaces in SAMPLE-KEY-NUMB for a START, STARTBR, or POINT call, and changes the value to 0.

Review the sample assembler exit PACKDEC, which is an example of a potential solution for a packed decimal field that is part of a key. Mapping SAMPLE-KEY-NUMB to a CHAR column is another solution.

The same issue potentially exists if your program specifies a key length that is less than the entire key.

Data conversion considerations

Unlike VSAM, DB2 enforces data value rules according to the data type of the column.

When your DB2 table includes column types of DECIMAL, INTEGER, SMALLINT, DATE, or TIME, a degree of data cleansing might be required regardless of whether you use CICS VT re-engineering or not. This is illustrated by examining how certain fields in the data set copybook shown in Figure 9 on page 26 are mapped.

Copybook field	Picture
ITEM-COST	S99999V99 COMP-3
ITEM-REORDER-QUANTITY	S999

Copybook field	Picture
ITEM-FIRST-SHIP-DATE	S(9) COMP-3

These fields are mapped to the following columns:

DB2 column	Data type
ITEM_COST	DEC(7,2)
ITEM_REORDER_NO	SMALLINT
ITEM_DATE_FSHIP	DATE

CICS VT automatically converts the data values in these three fields. If any record in either the field ITEM-REORDER-QUANTITY or ITEM-COST contains invalid data, the VIDLOAD program will abend with an S0C7 in the CICS VT module VIDCONV. The offending field value can be identified from the dump produced by the S0C7 abend as follows:

- Register 0 contains the address of the invalid field value.
- Register 8 contains the address of the CICS VT VIDFBEP control block. The CICS VT field name is four fullwords down VIDFBEP.

The format of the VIDFBEP control block is explained in the *CICS VT for z/OS Data Reengineering and Customization Guide*. Invalid data for the ITEM-FIRST-SHIP-DATE field does not result in an S0C7 abend, but causes an error in the DB2 LOAD utility. For example, a value of x'000999999C' or spaces cannot be converted into valid DB2 DATE column values, and will produce the following error:

```
DSNU334I -DB2B DSNURWBG - INPUT FIELD 'col-name' INVALID
         FOR 'table-name', ERROR CODE '14'
```

Avoiding data translation errors

Certain data conversion problems can be avoided by analyzing the data unloaded from VSAM before attempting to convert it with the VIDLOAD utility program.

If you add the DD statement VIDS0C7 to the JCL for the VIDLOAD utility program, CICS VT reads the data set produced by the VIDUNLOD utility program and identifies every field that would otherwise cause an S0C7 abend during data conversion. This is shown in the following JCL:

```
//VIDLOAD   EXEC PGM=VIDLOAD,PARM='dimname'
//STEPLIB   DD DSN=appl.DRIVERS.LOAD,DISP=SHR
//          DD DSN=VID.SVIDLODE,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//LOADIN    DD DSN=vsam.unload,DISP=SHR
//LOADOUT   DD DUMMY
//VIDS0C7   DD SYSOUT=*
```

CICS VT reads the entire file specified by the LOADIN DD statement and invalid records are written to the DD VIDS0C7. An example of the output produced by CICS VT follows:

```
VID DATA EXCEPTION REPORT
RECORD  0000000002 OFFSET  0029 DATASET VIDKSDS
        FIELD NAME VIDF004 LENGTH  0004 VALUE xxxx
RECORD  0000000217 OFFSET  0039 DATASET VIDKSDS
        FIELD NAME VIDF007 LENGTH  0005 VALUE yyyyy
```

These errors refer to the CICS VT sample file VIDKSDS. The fields in error, VIDF004 and VIDF007, are packed decimal fields, and the field values shown are invalid for this data type. If your DB2 table includes decimal or integer column

types, you should run the VIDLOAD utility with the VIDS0C7 DD statement before attempting to convert the VSAM data.

Variable length records in VSAM

Unless the last column in the DB2 table is VARCHAR, CICS VT assumes that a file contains fixed length records. This may impose restrictions on your tabledesign.

There are two typical cases where variable length records are used in VSAM:

- Your file contains free-format text data that is entered by a user. Your application programs limit the maximum size of the variable length field and programmatically set the length.
- Your file contains an array or repeating field or group, and you supply the length of each record as a program variable in a WRITE/REWRITE call. You may be using the OCCURS DEPENDING ON clause in COBOL.

In all cases, the VARCHAR column size must be the difference between the shortest and longest VSAM records.

Identifying varying record lengths

Any VSAM file can contain variable length records. If MAXLRECL is greater than AVGLRECL when you list the base VSAM cluster, your file might contain varying length records. However, if MAXLRECL equals AVGLRECL the file can contain variable length records. The only reliable way to establish that a file contains varying length records is to review the programs that insert new records.

Auto-mapping implications

There are two situations where the auto-mapper assumes that a file contains varying length records:

- If MAXRECL is not equal to AVGRECL and the last field in the copybook is not numeric, the last field is mapped to a VARCHAR column. If the file records are genuinely variable length, it is possible that more than just the last field is part of the varying length. In this case you have to manually update the DDL to combine these fields into a single column, and manually update the mapping.
- If OCCURS DEPENDING ON is specified for the last field or group field, the auto-mapper combines the entire array into a single VARCHAR column.

The auto-mapper issues an appropriate status message when it encounters either of these situations, as shown in Figure 62 on page 93.

```
 -------------------- CICS VT: Edit columns in AAAV21 -------- Row 24 from 27
Command ===> _____________________________________________ Scroll ===> CSR

Commands: MAP  SAVE  PREVIEW  SUSPEND/RESUME  CHANGE/UNDO  SHOW
Actions:  S Display, U Update, D Delete, I Insert            Status message /
A Pos   Copybook field name / DB2 column name     Type Len   Exit      Pic Par
- ----- ----+----1----+----2----+----3----+----   ---- ----- -----------------
_ 129   ZONED-RIPE-FOR-DEC-A                    +      6
        ZONED_RIPE_FOR_DEC_A.........           +  DEC 6,0
_ 135   ZONED-RIPE-FOR-DEC-B                    +      6
        ZONED_RIPE_FOR_DEC_B.........           +  DEC 6,2
_ 141   ZONED-NOT-RIPE-FOR-DEC                  +      8
        ZONED_NOT_RIPE_FOR_DEC........          + CHAR 8
_ 149   VARCHAR-ME-UP                           +      565   Assumed varying
        VARCHAR_ME_UP.................          + VARC 565
****************************** Bottom of data ******************************
```

Figure 62. Identifying potential varying length records

There are no issues if a file containing fixed length records is mapped to a DB2 table with VARCHAR as the last column. The reverse situation almost inevitably causes problems in your application.

Variable record lengths in CICS VT

The variable length field should always be the last field in a VSAM record. Similarly, a variable length column should always be the last column in the DB2 table. On a retrieval type call, CICS VT adds the VARCHAR column length from DB2 to the length of the record preceding the VARCHAR column, and returns this length to your application program. For PUT, WRITE,or REWRITE, CICS VT gets the record length from your program, subtracts the length of the fixed part of the record, and uses the remainder as the VARCHAR column length.

If no VARCHAR column exists in the table, CICS VT assumes the VSAM record is fixed length.

The VARCHAR column must be able to contain the entire variable area. This means that the difference in size between the shortest and the longest records in the file determines the VARCHAR column size.

DB2 table design example 1

Consider the copybook shown in Figure 63.

```
01  EXAMPLE-1-REC.
    05  EXAMPLE-1-KEY.
        10  EXAMPLE-1-APPL-NUMB  PIC X(12).
        10  EXAMPLE-1-BUREAU     PIC X(2).
        10  EXAMPLE-1-VIEW-NUMB  PIC 9(2).
        10  EXAMPLE-1-SEG.
            15 EXAMPLE-1-SEG-2   PIC X(2).
            15 EXAMPLE-1-SEG-4   PIC X(2).
        10  EXAMPLE-1-SEG-SEQ    PIC 9(4).
        10  EXAMPLE-1-SEG-SEQ-X REDEFINES
            EXAMPLE-1-SEG-SEQ    PIC X(4).
    05  EXAMPLE-1-DATA           PIC X(1000).
```

Figure 63. Variable record copybook example 1

The maximum record length for the file using this copybook is 1024 bytes and the field EXAMPLE-1-DATA contains between 0 and 1000 bytes of data. This means

that the shortest record length is 24 bytes. The record length of the DIM that is specified in the mapping for a variable length record file is always the maximum length, which is 1024 bytes in this case. The correct mapping for this copybook is to map the last 1000 bytes (for EXAMPLE-1-DATA) as a single field mapped to a DB2 column defined as VARCHAR(1000). The DDL is shown in Figure 64 .

```
CREATE TABLE VARIABLE_EXAMPLE1(
    EXAMPLE1_APPL_NUMB    CHAR(12) NOT NULL
   ,EXAMPLE1_BUREAU       CHAR(02) NOT NULL
   ,EXAMPLE1_VIEW_NUMB    DEC(2,0) NOT NULL
   ,EXAMPLE1_SEG_2        CHAR(02) NOT NULL
   ,EXAMPLE1_SEG_4        CHAR(02) NOT NULL
   ,EXAMPLE1_SEG_SEQ      CHAR(04) NOT NULL
   ,EXAMPLE1_DATA         VARCHAR(1000)
   ,PRIMARY KEY (EXAMPLE_APPL_NUMB));
```

Figure 64. Variable record DDL example 1

DB2 table design example 2

When variable length records occur because of an array, the solution is similar to example 1. The last column in the DB2 table must be VARCHAR and must contain the entire variable area.

In the copybook shown in Figure 65, the auto-mapper is processing it as varying length because MAXLRECL is not equal to AVGLRECL. There is no DEPENDING ON clause for the array EXAMPLE-2-DEBT-INFO in this situation.

```
01  EXAMPLE-2-REC.
    05  EXAMPLE-2-KEY.
        10  EXAMPLE-2-APP-NUM              PIC 9(12).
        10  EXAMPLE-2-SEQ-NUM              PIC 9(01).
    05  EXAMPLE-2-COMMON-DATA.
        10  EXAMPLE-2-UNIFIED-CR-TRAN-ID   PIC X(18).
        10  EXAMPLE-2-EFX-PERSISTENT-KEY   PIC X(12).
        10  EXAMPLE-2-CREATE-DATE          PIC 9(08).
        10  EXAMPLE-2-CREATE-TIME          PIC 9(06).
    05  EXAMPLE-2-DEBT-INFO OCCURS 20 TIMES.
        10  EXAMPLE-2-PAYMENT-AMOUNT       PIC S9(09)V99.
        10  EXAMPLE-2-PAYMENT-CODE         PIC X(01).
        10  EXAMPLE-2-AFFILIATE-TYPE       PIC X(01).
        10  EXAMPLE-2-AFFILIATE-ID         PIC X(02).
        10  EXAMPLE-2-ACCOUNT-NUMBER       PIC X(14).
        10  EXAMPLE-2-MARKET               PIC X(03).
```

Figure 65. Variable record copybook example 2

The maximum record length for the file using this copybook is 697 bytes. The fixed length portion of the copybook is 57 bytes and the group field EXAMPLE-2-DEBT-INFO is 32 bytes. Because the group field occurs 20 times, the total length of the array is 640 bytes.

Each record in the VSAM file contains between 0 and 20 instances of the group field, controlled by the application. This entire area must be mapped to a single DB2 column defined as VARCHAR(640). This is shown in Figure 66 on page 95.

```
CREATE TABLE VARIABLE_EXAMPLE2(
    EXAMPLE2_APPL_NUMB            CHAR(12) NOT NULL
   ,EXAMPLE2_SEQ_NUM              CHAR(01) NOT NULL
   ,EXAMPLE2_UNIFIED_CR_TRAN_ID   CHAR(18) NOT NULL
   ,EXAMPLE2_EFX_PERSISTENT_KEY   CHAR(12) NOT NULL
   ,EXAMPLE2_CREATE_DATE          DATE     NOT NULL
   ,EXAMPLE2_CREATE_TIME          TIME     NOT NULL
   ,EXAMPLE2_DEBT_INFO            VARCHAR(640) NOT NULL
   ,PRIMARY KEY (EXAMPLE_APPL_NUMB, EXAMPLE2_SEQ_NUM));
```

Figure 66. Variable record DDL example 2

Defining variable length files in CICS

The CICS VT DST entry for files mapped as variable length must have `RECORDF=VAR` specified. Otherwise a LENGERR error is returned to your application.

Conversion utility for mapping to multiple DB2 tables

VSAM data sets that include multiple record types can be mapped to multiple DB2 tables. CICS VT provides a facility to do this.

A combination of FBE and IRD exits is required and these are covered in the *CICS VT Data Reengineering and Customization Guide*. When you implement this type of solution, the data migration process has to convert the VSAM data set records into multiple record types for loading into multiple DB2 tables.

A slightly modified version of the VIDLOAD utility shown in Figure 58 on page 85 is required. The only difference is that you need to change the LOADOUT DD statement to LOADOUTM.

The record length of the output data set must be the length of the longest DB2 row plus an additional two bytes. The VIDLOAD conversion program, in conjunction with functionality in the IRD exit, uses these two bytes to identify the appropriate record type. Use this record type field as the WHEN condition in the DB2 LOAD utility control cards as follows:

```
WHEN (1:02) = 'A1'
```

Chapter 4. Enabling batch access to a migrated data set

Before you can start testing your application programs, you must enable access to the migrated VSAM data set. For batch application programs, you need to change the JCL. This section explains the mandatory and optional JCL changes.

You should also understand the operational aspects concerning CICS VT migration by reading Chapter 8, "CICS VT operational and recovery considerations," on page 125.

Batch JCL changes can be made in advance of migrating a VSAM file. This is explained in "Making JCL changes before migration" on page 99.

STEPLIB or JOBLIB changes

The CICS load library must be added to either the STEPLIB or the JOBLIB DD statement or the system linklist.

The application driver library where you generate the DIM and DDM driver modules and the CICS VT system library must be available at run time. This means that you must add them to the STEPLIB or JOBLIB DD statement, or add them to the linklist.

Here are sample JCL cards:

```
//STEPLIB  DD DSN=appl.prog.lib,DISP=SHR
//         DD DSN=appl.drivers.load,DISP=SHR
//         DD DSN=VID.SVIDLODE,DISP=SHR
```

You should place the CICS VT drivers library ahead of the VID system library.

VSAM file DD statement changes

The method that CICS VT uses to intercept VSAM calls from your batch application programs is explained in "The runtime component" on page 26. To enable call interception, update the DD statements that define the VSAM base cluster and alternate index data sets. Below is the DD statement before and after CICS VT migration for the KSDS data set used earlier in this chapter:

Before:

```
//KSDS001  DD DSN=VID.VIDKSDS,DISP=SHR
//KSDSAIX  DD DSN=VID.VIDKSDS.AIX,DISP=SHR
```

After:

```
//KSDS001  DD SUBSYS=(ssi,db2id,DIM name)
//KSDSAIX  DD SUBSYS=(ssi,db2id,AIX DIM name)
```

The parameter values for the SUBSYS statement are:

Parameter	Description
ssi	The name of the VSAM subsystem interface that you specified at CICS VT installation. See "Installing the batch subsystem" on page 12.

Parameter	Description
db2id	The subsystem ID of the DB2 table containing the migrated data.
DIM name	The name of the DIM driver. This is also the default name of the plan used by CICS VT to connect to DB2. If you want to use the read-only DDM, specify the name of the read-only DDM module instead of the DIM.
AIX® DIM name	The name of the DIM driver. This is also the default name of the plan used by CICS VT to connect to DB2.

You can override the default plan name. See " Overriding the default DB2 plan name " on page 128. Ensure that the batch job user ID has the required authority to execute the DB2 plan.

Additional DD statement for dummy VSAM data set

PL/I and COBOL use undocumented interfaces to invoke the VSAM SHOWCB and TESTCB macros. To provide this support in CICS VT, you must create a new VSAM data set and add one additional DD statement for each base cluster and alternate index path to your batch JCL.

If you use the data migration generation utility, run the member VID4DUMM in the generated *appl.dimname.migrate* library to create the dummy VSAM data set. If you don't use the data migration generation facility, you must create the dummy data set manually.

Model the dummy data set cluster on the cluster for the original VSAM data set, with minimum space allocated. Figure 67 is an example data set that uses the definition for the cluster that was used in "The mapping component" on page 26 .

```
DEFINE  CLUSTER
          (NAME(VID.VIDKSDS.VID)
           INDEXED SHR(2 3))
        DATA
          (NAME(VID.VIDKSDS.VID.DATA)
           KEYS(6 0) VOL(CTC002) RECSZ(80 80)
           CISZ(16384) TRK(1))
        INDEX
          (NAME(VID.VIDKSDS.VID.INDEX)
           CISZ(2048) VOL(CTC002) TRK(1))
```

Figure 67. Define CLUSTER for dummy VSAM data set

Initialize the dummy data set with one record. Sample JCL using the Access Method Services IDCAMS utility follows:

```
//VIDCOPY EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD  SYSOUT=*
//INF      DD  DISP=SHR,DSN=VID.VIDKSDS
//OUTF     DD  DISP=SHR,DSN=VID.VIDKSDS.VID
//SYSIN    DD  *
 REPRO INFILE(INF) OUTFILE(OUTF) COUNT(1)
```

You must add a new DD statement for this copy of the VSAM data set to your batch JCL for the base cluster data set and each alternate index. Use the dummy VSAM data set name for both the base cluster statement and alternate index path DD statements. The DD name you must use is derived from the DD statement for your real VSAM data set as follows:

- If the DD statement for your existing VSAM data set is 7 characters or less, add an at sign (@) character as a suffix. For example, if the DD statement is ITEMDET, then the new DD statement to add is ITEMDET@.
- If the DD statement for your existing VSAM data set is 8 characters, the at sign replaces the first character. For example, if the DD statement is ORDERDET, then the new DD statement to add is @RDERDET.

Examples of both DD statements are shown in the following sample JCL:

```
//STEPLIB  DD DSN=appl.prog.lib,DISP=SHR
//         DD DSN=appl.DRIVERS.LOAD,DISP=SHR
//         DD DSN=VID.SVIDLODE,DISP=SHR
//ITEMDET  DD SUBSYS=(ssi,db2id,dimname)
//ITEMDET@ DD DSN=ITEMDET.VID.DUMMY,DISP=SHR
//ITEMDETX DD SUBSYS=(ssi,db2id,aix-dimname)
//@TEMDETX DD DSN=ITEMDET.VID.DUMMY,DISP=SHR
//ORDERDET DD SUBSYS=(ssi,db2id,dimname)
//@RDERDET DD DSN=ORDERDET.VID.DUMMY,DISP=SHR
```

In this example, ITEMDET is the DD statement for the base cluster, and ITEMDETX is the DD statement for an alternate index path. The dummy VSAM data set should be defined to CICS. This is covered in "Disabling access to a VSAM data set" on page 110.

You can choose your own rules for the dummy VSAM data set DD name by adding appropriate code to the CICS VT user exit VIDDDMEX.

CICS VT tracing

Additional DD statements are required to enable CICS VT tracing. The trace facility is optional and is useful during testing, especially if you are performing data re-engineering.

See "Activating a trace in batch jobs" on page 139 for instructions on how to enable CICS VT tracing in a batch job.

Making JCL changes before migration

It is possible to make the required batch JCL changes in advance of migrating the VSAM data to DB2, so that your application programs continue to access the VSAM data set. You direct CICS VT to access the data either in VSAM or in DB2 through the user-controlled program VIDFMSPX.

When you implement the required changes, the final JCL to enable CICS VT looks like this:

```
//jobcard
//STEPLIB  DD DSN=appl.prog.lib,DISP=SHR
//         DD DSN=appl.DRIVERS.LOAD,DISP=SHR
//         DD DSN=VID.SVIDLODE,DISP=SHR
//ITEMDT   DD SUBSYS=(ssi,db2id,vtitem)
//ITEMDT@  DD DSN=ITEMDT.VID.DUMMY,DISP=SHR
//ORDERDT  DD SUBSYS=(ssi,db2id,vtorder)
//ORDERDT@ DD DSN=ORDERDT.VID.DUMMY,DISP=SHR
//AUDIT    DD SUBSYS=(ssi,db2id,vtaudit)
//AUDIT@   DD DSN=AUDIT.VID.DUMMY,DISP=SHR
//AUDAIX   DD SUBSYS=(ssi,db2id,vtaudix)
//AUDAIX@  DD DSN=AUDIT.VID.DUMMY,DISP=SHR
```

VIDFMSPX is a user-controlled data-only module containing one or more DIM names. It operates like a switch. If you update your batch JCL before migrating the VSAM data set but want to continue to access the VSAM data, add the DIM name to the VIDFMSPX source and assemble and link it in any JOBLIB or STEPLIB data set. There is sample VIDFMSPX code and assemble/link JCL in *my*.SVIDCNFG.*custom* member VIDGFMX.

By making the required JCL changes in advance, you reduce the time and effort to switch to DB2 access after the data is migrated. Instead of having to make the JCL changes at the same time as you migrate the VSAM data, you can separate migration preparation activities, such as JCL modification, from data migration activities.

The VIDFMSPX module

When you add the SUBSYS statement and CICS VT parameters to an existing VSAM DD statement, all application calls to that DD are processed by the CICS VT subsystem. The first VSAM call in a batch application program opens the data set. If there is no DSN parameter, the subsystem assumes that the data has been migrated to DB2. If the DSN parameter exists, the subsystem attempts to load VIDFMSPX from the STEPLIB or JOBLIB libraries. If the module doesn't exist, CICS VT processes the appropriate VSAM call in DB2. If VIDFMSPX is successfully loaded, VT scans the module for the DIM name specified 3rd SUBSYS parameter. If the DIM is defined in VIDFMSPX, all calls to this DD statement are processed by VSAM. If the DIM is not defined in VIDFMSPX, all calls to this DD statement are processed by CICS VT.

For example, assume that you are using CICS VT to migrate your Order Detail, Item Detail, and Audit files. Figure 68 shows the JCL changes that prepare for CICS VT migration but still allow the VSAM file to be accessed as normal:

```
//jobcard
//STEPLIB  DD DSN=appl.prog.lib,DISP=SHR
//         DD DSN=appl.DRIVERS.LOAD,DISP=SHR
//         DD DSN=VID.SVIDLODE,DISP=SHR
//ITEMDT   DD SUBSYS=(ssi,db2id,vtitem),
//         DISP=SHR,DSN=ITEMDT.VSAM
//ITEMDT@  DD DSN=ITEMDT.VID.DUMMY,DISP=SHR
//ORDERDT  DD SUBSYS=(ssi,db2id,vtorder),
//         DISP=SHR,DSN=ORDERDT.VSAM
//ORDERDT@ DD DSN=ORDERDT.VID.DUMMY,DISP=SHR
//AUDIT    DD SUBSYS=(ssi,db2id,vtaudit),
//         DISP=SHR,DSN=AUDIT.VSAM
//AUDIT@   DD DSN=AUDIT.VID.DUMMY,DISP=SHR
//AUDAIX   DD SUBSYS=(ssi,db2id,vtaudix),
//         DISP=SHR,DSN=AUDIT.VSAM.PATH
//AUDAIX@  DD DSN=AUDIT.VID.DUMMY,DISP=SHR
```

Figure 68. JCL changes before CICS VT migration

Assuming that the DIM names ar VTITEM, VTORDER, VTAUDIT, and VTAUDIX, the VIDFMSPX source code that you have to assemble and link to continue to access the VSAM data is as follows:

```
VIDFMSPX CSECT
VIDFMSPX AMODE 31
VIDFMSPX RMODE ANY
VERIFY   DC    CL8'VIDFMSPX'
*
*        SPECIFY DIM NAMES HERE
         DC    CL8'VTITEM'
```

```
         DC    CL8'VTORDER'
         DC    CL8'VTAUDIT'
         DC    CL8'VTAUDIX'
THEEND   DC    XL8'FFFFFFFFFFFFFFFF'
         END
```

The CICS VT subsystem issues the following message when calls are redirected to VSAM:

```
+VIDSS250 SUBSYSTEM NAME=ssid VSAM ONLY FOR DDNAME=ddname
```

Managing the VIDFMSPX module

VIDFMSPX provides the flexibility to perform a staged migration process. If you exploit this capability, you must implement a mechanism to control VIDFMSPX.

You should assemble and link VIDFMSPX into your CICS VT application drivers library. If you are deploying a single, system-wide application drivers library, you will have a single instance of VIDFMSPX and therefore you require a single p oint of control. If you have separate drivers libraries on an application by application basis, you can have multiple application level instances of VIDFMSPX.

When you remove a DIM from VIDFMSPX, you should also remove the JCL references to the original VSAM data set.

Chapter 5. Enabling CICS access to a migrated data set

After a data set has been migrated to DB2, you can enable access from CICS programs.

The first step is to add an entry for the data set to the CICS VT data set table (DST). Then define the CICS VT drivers and any CICS VT exits to CICS, before running the VTMI CICS transaction to introduce the newly assembled DST. These steps are described in detail in the following section.

Entries in the DST with MSTATUS=INACT are not considered to be migrated to DB2. You use the VTMM transaction to switch the status between INACT and ACT. The status can be changed at any time, and persists while CICS is active. The VTMI RESTART COLD transaction reloads the DST from disk and resets the migration status to its definition in the DST.

You must add a DB2ENTRY entry for the transactions that access the migrated data set to enable the connection to DB2. This is explained in "CICS DB2ENTRY entries" on page 108.

You should understand the operational aspects concerned with CICS VT migration by reading Chapter 8, "CICS VT operational and recovery considerations," on page 125.

General CICS file control considerations

When a VSAM file is added to the CICS VT DST and the migration status is `ACT`, the VSAM base cluster associated with the file control entry is no longer accessed. There are a number of operational factors to be considered.

- If you routinely enable/disable or stop/start CICS files and want to continue to exercise similar control after DB2 migration, review "Controlling migrated data sets in CICS" on page 108.
- If you want to take steps to prevent accidental and potentially undetected access to the real VSAM cluster, review "Disabling access to a VSAM data set" on page 110.
- If the file is defined to a FOR, review "MRO considerations" on page 110.

The CICS file control definition must exist post migration to DB2. The following file control attributes are no longer relevant after migration:

- RLS
- NSR/LSR buffers

The following file control attributes continue to be relevant after migration:

- Opentime
- Enablestatus
- Readstatus
- Updatestatus
- Addstatus
- Browsestatus
- Deletestatus

Existing files defined as `Opentime Startup` should be changed to `Opentime Firstref`.

The CICS VT global user exit (GLUE) honours these status attributes if they are dynamically changed by, for example, the CEMT transaction or CICS Explorer®.

If you migrate files that are defined to CICS as user maintained data tables (UMDT) or CICS maintained data tables (CMDT), modify the file attributes to `TABLE - No` at the same time as you add it to the CICS DST. Potentially, the CICS VT GLUE may not be enabled sufficiently early in the CICS start-up process to handle data tables. To avoid potential performance bottlenecks associated with data tables, consider defining DB2 buffer pools exclusively for the appropriate tables.

Adding a new data set to the CICS VT data set table

The CICS VT DST module name is VIDCDTAB. A version containing a definition for the sample data set VIDKSDS is available in VID.SVIDLODE. The macro source for this module is included as member VIDCDTAB in VID.SVIDCNFG. The JCL to assemble the macro source DST is member VIDGDTAB in *my*.SVIDCNFG.*custom*.

If you use the data migration generation utility, the member VID9DST in the generated *appl.dimname.migrate* library has the appropriate macro statements to add the base cluster DIM to the DST. You must code the macro statements manually for alternate index DIMs. If you don't use the data migration generation facility, you must code the DIM and alternate index macro statements manually.

The VIDCTAB source contains multiple occurrences of the CICS VT macro VIDTAB. Each occurrence is qualified with a `TYPE` parameter specifying one of three potential values:

- The first occurrence of the VIDTAB macro must be `TYPE=INITIAL`.
- There are multiple VIDTAB occurrences of `TYPE=ENTRY`, one for each migrated VSAM file.
- The last occurrence of the VIDTAB macro must be `TYPE=FINAL`.

Sample DST source can be found in *my*.SVIDSAMP.*custom* member VIDCDTAB.

Add a TYPE=ENTRY statement for each migrated data set to enable access from CICS programs. The parameters for the TYPE=ENTRY statement are as follows:

FILE= *filename*
The name used to define the file to CICS in the CSD.

DIM= *dimname*
The name of the CICS VT DIM module. This can be the same name as the **FILE** parameter value.

STATUS= ENA|DIS|UNE
The initial status of the file. Possible values are as follows:

ENA The file is ENABLED. The file can be opened using a **SET FILE OPEN** command, or implicitly by CICS when it is next referenced.

DIS The file is DISABLED. The file cannot be opened unless it is first enabled.

UNE The file is UNENABLED. The file cannot be opened implicitly by CICS when it is next referenced. It can be opened by issuing the **SET FILE OPEN** command.

See the *CICS Resource Definition Guide* for an explanation of these parameter values.

MSTATUS= ACT|INACT
The file migration status. Possible values are as follows:

ACT The file is migrated and data access is DB2.

INACT
The file is not migrated and data access is VSAM.

OPEN= S|F
The time when the file is opened. Possible values are as follows:

S The file is opened at CICS startup.

F The file is opened at first access.

ADD= NO|YES
Whether new records can be added or not.

BRO= NO|YES
Whether the file can be browsed.

DEL= NO|YES
Whether records can be deleted.

REA= NO|YES
Whether the file can be read.

UPD= NO|YES
Whether the file can be updated.

RESET= NO|YES
If the VSAM file is defined with the REUSE attribute, specify `YES`. Otherwise specify `NO`.

RECORDF= FIX|VAR
The format of the record. Possible values are as follows:

FIX Fixed length **EXEC CICS** file control calls.

VAR Variable length **EXEC CICS** file control calls. Specify `RECORDF=VAR` if you mapped the file as varying length otherwise a LENGERR error is returned when the file is accessed by your application.

ERRORACT= ABEND |REPORT
The action to take if DMF detects a difference. Possible values are as follows:

ABEND
The transaction abends. If the data area is different, the abend code is 3139. If the return code or reason code is different, the abend is 3141.

REPORT
Error details are written to the VIDCOMP DD statement and the transaction continues.

DUMP= NO |YES
When the **ERRORACT** parameter value is `ABEND` , use this parameter to indicate if a dump is required.

Model the parameter values for the CICS VT DST entry on the parameters of the file as it is defined in CICS .

After you have added a new entry, assemble the CICS VT DST with the VIDGDTAB member into a library that is part of the CICS DFHRPL concatenation. Ideally, this should be the same library that you use for your CICS VT driver modules. You must add entries for the base cluster and alternate index data sets.

Managing VIDCDTAB in a test environment

You can add all files that you are migrating with CICS VT to the DST at the same time, with the initial migration status parameter MSTATUS set to `INACT`. During application testing, use the VTMM transaction to switch between native VSAM access and CICS VT access.

The migration status set by the VTMM transaction persists until any of the following events:

- It is reset by VTMM *file-name* ACT or INACT
- It is reset by VTMI RESTART COLD
- CICS is stopped and restarted

The VTMM and VTMI transactions are explained in detail in "Additional CICS transactions" on page 19.

After you have completed testing for a file or group of files, update the DST MSTATUS parameter for these files to `ACTIVE` and assemble a new DST.

Defining CICS VT driver modules in CICS

Define the CICS VT DIM and DDM driver modules in CICS. If you use the data migration generation utility, the member VID9CSD in the generated *appl.dimname.migrate* library contains the input to the batch CSD update for a specific DIM and DDM. If you add the definitions manually, the attributes of all CICS VT driver modules should match those shown in Figure 69 on page 107.

```
 CHARACTERISTICS                              CICS RELEASE = 0620
  CEDA  View PROGram( VIDKSDS  )
  PROGram        : VIDKSDS
  Group          : VID
  DEscription    : VID DIM Moduule
  Language       : Assembler        CObol | Assembler | Le370 | C | Pli
  RELoad         : No               No | Yes
  RESident       : No               No | Yes
  USAge          : Normal           Normal | Transient
  USElpacopy     : No               No | Yes
  Status         : Enabled          Enabled | Disabled
  RSl            : 00               0-24 | Public
  CEdf           : Yes              Yes | No
  DAtalocation   : Any              Below | Any
  EXECKey        : User             User | Cics
  COncurrency    : Threadsafe       Quasirent | Threadsafe
 REMOTE ATTRIBUTES
  DYnamic        : No               No | Yes
+ REMOTESystem   :

                                                    SYSID=MICA
APPLID=A06CICS

 PF 1 HELP 2 COM 3 END     6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

Figure 69. CICS VT driver attributes in CICS

You should define CICS VT driver modules using the batch CSD update program. This allows you to port the definitions to other CICS systems, for system testing or production cutover. You should also remember that when a CICS transaction accesses a CICS VT migrated data set, DB2 access takes place and the transaction must be defined accordingly.

If required, you can define CICS VT driver programs to CICS as RESIDENT.

Defining CICS VT user exits in CICS

CICS VT user exits must be defined to CICS. Exits written in assembler must be coded to be REENTRANT. Consider using the CICS batch CSD update utility for portability between system.

If you use the data migration generation utility, the member VID9CSD in the generated *appl.dimname.migrate* library contains the input to the batch CSD update for FBEs and IRDs.

Activating a new DST

Use the VTMI supplied transaction to control the interface between CICS and CICS VT. To activate a new CICS VT DST, run the VTMI transaction using the following command:

```
VTMI RESTART option
```

There are two possible values for *option* as follows:

- If you specify the value *WARM*, a new DST is loaded and the migration status set by the previous occurrences of the VTMM transaction persists.
- If you specify the value *COLD*, a new DST is loaded and the migration status of every file is reset to the migration status defined in the DST.

The message `VIDCINIT - CICS VT successfully restarted` is issued indicating that the new CICS VT DST is active.

CICS VT tracing

CICS VT tracing provides diagnostic information to help you perform problem determination and test your user exits. You can also use tracing to confirm that CICS VT is accessing a migrated data set. See “CICS VT tracing” on page 137 for instructions on enabling and activating the CICS VT trace facility in CICS.

CICS DB2ENTRY entries

CICS transactions that access a migrated data set must be able to establish a connection to DB2.

The appropriate RDO definitions must be added for each transaction to provide the name of the DB2 plan to be used. This might require a combination of DB2CONN, DB2ENTRY, and DB2TRAN definitions, based on your local CICS and DB2 standards.

Controlling migrated data sets in CICS

The CICS VT GLUE is called VIDCGLUE, and is automatically invoked every time a migrated data set is processed. Calls initiated from CICS terminals such as CEMT or CECI are processed by VIDCGLUE.

Calls from application programs that use the systems programming interface SET FILE or SET DATA SET are also processed by VIDCGLUE. This means that CICS file control calls have the same effect in CICS VT as they do in CICS. For example, here is the response to a CEMT transaction inquiring on the status of the file KSDS05:

```
I FILE(KSDS05)
 STATUS:  RESULTS - OVERTYPE TO MODIFY
 Fil(VIDKSDS ) Vsa Ope Ena Rea Upd Add Bro Del     Sha
        Dsn(VID.KSDS05                               )
```

Figure 70. CICS CEMT inquire file results

The CICS VT VTMD supplied transaction, described in “VTMD - displaying the CICS VT DST” on page 19, displays the status of migrated files in CICS. The response to the VTMD transaction is shown in Figure 71 on page 109:

```
        08/31/11     ==>  CICS VT File Definitions  <==     16:24:22

Filename  DIM name   Org   Status            MStatus Operations    Compare
KSDS01    KSDS01           ENA INI           ACT     R U A B D     REP   N
KSDS02    KSDS02           ENA INI           ACT     R U A B D     REP   N
KSDS02P   KSDS02AI         ENA INI           ACT     R U A B D     REP   N
KSDS03    KSDS03           ENA INI           ACT     R U A B D     REP   N
KSDS03P1  KSDS03P1         ENA INI           ACT     R U A B D     REP   N
KSDS04    KSDS04           ENA INI           ACT     R U A B D     REP   N
KSDS05    KSDS05           ENA INI           ACT     R U A B D     REP   N
KSDS05P1  KSDS05P1         ENA INI           ACT     R U A B D     REP   N
KSDS05P2  KSDS05P2         ENA INI           ACT     R U A B D     REP   N
KSDS05P3  KSDS05P3         ENA INI           ACT     R U A B D     REP   N
KSDS05P4  KSDS05P4         ENA INI           ACT     R U A B D     REP   N
KSDS06    KSDS06           ENA INI           ACT     R U A B D     REP   N
KSDS06P1  KSDS06P1         ENA INI           ACT     R U A B D     REP   N
KSDS07    KSDS07           ENA INI           ACT     R U A B D     REP   N
KSDS07P1  KSDS07P1         ENA INI           ACT     R U A B D     REP   N
KSDS08    KSDS08           ENA INI           ACT     R U A B D     REP   N
KSDS09    KSDS09           ENA INI           ACT     R U A B D     REP   N
KSDS10    KSDS10           ENA INI           ACT     R U A B D     REP   N
More
PF3=end PF7=back PF8=fwd                          SYSID=S650 APPLID=CICSTS32
```

Figure 71. CICS VTMD file definition list screen

The status of the file according to the CEMT transaction is always consistent with the file status according to CICS VT, because CICS VT intercepts CEMT and CECI transactions and updates the CICS VT file status when necessary. This means that you have the same control over migrated files as you do with non-migrated files. For example, assume that you issue the CICS command to close the data set:

```
CEMT SET FI(KSDS05) Clo
```

When you run the transaction VTMD, you get the following results:

```
        08/31/11     ==>  CICS VT File Definitions  <==     16:24:45

Filename  DIM name   Org   Status            MStatus Operations    Compare
KSDS01    KSDS01           ENA INI           ACT     R U A B D     REP   N
KSDS02    KSDS02           ENA INI           ACT     R U A B D     REP   N
KSDS02P   KSDS02AI         ENA INI           ACT     R U A B D     REP   N
KSDS03    KSDS03           ENA INI           ACT     R U A B D     REP   N
KSDS03P1  KSDS03P1         ENA INI           ACT     R U A B D     REP   N
KSDS04    KSDS04           ENA INI           ACT     R U A B D     REP   N
KSDS05    KSDS05     KSDS  UNE               ACT     R U A B D     REP   N
KSDS05P1  KSDS05P1         ENA INI           ACT     R U A B D     REP   N
KSDS05P2  KSDS05P2         ENA INI           ACT     R U A B D     REP   N
KSDS05P3  KSDS05P3         ENA INI           ACT     R U A B D     REP   N
KSDS05P4  KSDS05P4         ENA INI           ACT     R U A B D     REP   N
KSDS06    KSDS06           ENA INI           ACT     R U A B D     REP   N
KSDS06P1  KSDS06P1         ENA INI           ACT     R U A B D     REP   N
KSDS07    KSDS07           ENA INI           ACT     R U A B D     REP   N
KSDS07P1  KSDS07P1         ENA INI           ACT     R U A B D     REP   N
KSDS08    KSDS08           ENA INI           ACT     R U A B D     REP   N
KSDS09    KSDS09           ENA INI           ACT     R U A B D     REP   N
KSDS10    KSDS10           ENA INI           ACT     R U A B D     REP   N
More
PF3=end PF7=back PF8=fwd                          SYSID=S650 APPLID=CICSTS32
```

Figure 72. VTMD transaction results

The file definition list screen shows that KSDS05 has been changed to `UNE`. This means that CICS VT will not process calls for this data set because it has been disabled.

To be able to change the status of a file from a CICS perspective, the VSAM data set associated with the file control entry must exist. You can associate any VSAM data set with the file control entry providing that the file is always accessed by CICS VT.

Disabling access to a VSAM data set

After a data set has been migrated to DB2, the VSAM data set is no longer accessed by CICS. Providing that the data set is defined in the CICS VT DST, CICS programs will access the data in DB2.

If the CICS VT interface in CICS is stopped for any reason or a file migration status is set to `INACT`, CICS attempts to process file control calls to this file against the actual VSAM data set. You should take steps to ensure that the original VSAM file cannot be accessed accidentally. For example, update the file control definition and specify a non-existent data set name. If you need to be able to stop and start the file, as discussed in "Controlling migrated data sets in CICS" on page 108, the data set name in the file control definition must exist. Do not use the dummy data set that is required for CICS VT batch programs.

MRO considerations

In a typical MRO environment, the application runs in an AOR and VSAM files are defined as remote files in a separate region such as an FOR. CICS VT supports access to remote VSAM files but there are operational implications to consider.

CICS VT checks that the file is available for processing. If the file is defined as remote, CICS VT is unable to check the file status and assumes it is available. If you need to be able to disable and enable access to a file that has been migrated, you must define the file in the same region as CICS VT.

There is an unnecessary path length overhead for remotely defined files that are migrated with CICS VT. Consider defining remote files as local to avoid function shipping. Remote files accessed from multiple AORs should be defined as local to each.

Disabling CICS VT from a CICS system

You can disable CICS VT from a CICS system using the following VTMI command:

```
VTMI STOP
```

CICS file control calls are no longer intercepted by the CICS VT GLUE.

Chapter 6. The dual mode facility (DMF)

The primary purpose of the dual mode facility (DMF) is to help identify errors in data set mapping, especially where re-engineering is achieved using user-written exits. This normally occurs during the data verification phase of data migration. You can also use DMF to support application testing.

When DMF is activated, VSAM calls from application programs are processed by CICS VT against both VSAM data and DB2 data. The data areas, return codes, and reason codes are automatically compared. If any difference is detected in a batch program, an abend occurs. If any difference is detected in a CICS program, the action taken by CICS VT depends on the value of the ERRORACT DST option.

The checking performed by DMF varies according to the type of VSAM call as follows:

- The data areas, return codes, and reason codes are compared for READ, BROWSE, and GET calls.
- The return codes and reason codes are compared for WRITE, REWRITE, and PUT calls.
- The return codes and reason codes are compared for ERASE calls. In addition, the actual count of records erased is also compared.

The return code and reason code are compared first, followed by the data areas. If DMF detects any difference, diagnostic data is produced and the application program is abnormally terminated.

You should not use DMF in a production capacity.

The following terminology is used in this chapter:

- The *original* VSAM data set refers to the data set as it existed in VSAM.
- The *migrated* VSAM data set refers to the file after it has been migrated to DB2.

DMF for CICS programs

There are two parts to enabling the DMF for CICS programs.

1. The original VSAM data set must be defined to CICS file control using a predefined file name. This can be defined as either local or remote.
2. The DMF must be activated at a CICS system level using the VTMA transaction.

The definition of the data set in the CICS VT data set table (DST) includes two optional parameters relating to DMF: **ERRORACT** and **DUMP**. These are discussed in "Adding a new data set to the CICS VT data set table" on page 104.

Output from the DMF is written to the CICS DD statement VIDCOMP.

If you are using MRO and have defined the DMF file as remote, CICS VT assumes that it is always available for processing.

Defining the VSAM data set to CICS

To compare the results of the each call in VSAM and DB2, you must add a new file control definition to CICS file control.

The name for the new file definition is derived from the existing file control definition as follows:

- If the file name is 7 characters or less, specify '@' as a suffix.
- If the file name is 8 characters, change the first character to '@'.

Figure 73 shows the response to the CEMT transaction illustrating both of these situations:

```
I FI(*)
STATUS:  RESULTS - OVERTYPE TO MODIFY
 Fil(@RDERFIL  Vsa Clo Ena Rea Upd Add Bro Del     Sha
       Dsn( VID.ORDERFIL                          )
 Fil(ORDERFIL) Vsa Ope Ena Rea Upd Add Bro Del     Sha
       Dsn( VID.ORDERFIL.DUMMY                    )
 Fil(VIDKSDS ) Vsa Ope Ena Rea Upd Add Bro Del     Sha
       Dsn( VID.VIDKSDS.DUMMY                     )
 Fil(VIDKSDS@) Vsa Ope Ena Rea Upd Add Bro Del     Sha
       Dsn( VID.VIDKSDS                           )
 Fil(VIDRRDS)  Vsa Ope Ena Rea Upd Add Bro Del     Sha
       Dsn( VID.VIDRRDS.DUMMY                     )









                                                SYSID=CICS APPLID=A06CICS1
  RESPONSE: NORMAL                          TIME:  12.50.29  DATE: 14.01.05
PF 1 HELP       3 END       5 VAR       7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 73. CEMT display of files defined to DMF

@RDERFIL and VIDKSDS@ are the new file control entries defining the original VSAM data sets that DMF uses for files ORDERFIL and VIDKSDS respectively. The file VIDRRDS is not eligible for DMF because there is no file control definition for the original VSAM data set.

Activating DMF with the VTMA transaction

DMF is activated at a CICS system level using the VTMA transaction.

To issue the transaction, use the command **VTMA START**. All files that have been defined to CICS using the DMF file names are processed by DMF. Using the example in Figure 73, DMF is active for all application programs accessing the files VIDKSDS and ORDERFIL.

Interoperability between VTMI and VTMA

You cannot start the DMF unless the CICS VT interface is active.

The CICS VT interface is normally started by the PLT program VIDCINIT. If you try to stop the CICS VT interface while the DMF is active, you receive the following message:

```
+VIDCINIT - CICS VT Compare is active. Stop using VTMA.
```

You must issue the **VTMA STOP** command before you can issue **VTMI STOP**.

Enabling DMF at CICS start up

If you need to use DMF for an extended period, it can be automatically enabled at CICS start up by adding the program VIDCINIC to the CICS PLT.

With VIDCINIC in the PLT, an additional CICS VT message is issued at CICS startup. The message is as follows:

```
+VIDCINIC - CICS VT DMF successfully initialised.
```

DMF error reporting in CICS

The DMF records information when it detects differences in the data.

```
**********************************************************************
* CICS VT COMPARE EXCEPTION                                          *
**********************************************************************

TRAN: K1P4  DIM: KSDS01 TERMINAL: 0019 TIME: 15.49.5300  REQ: 0000026

 EXEC CICS READNEXT
  FILE(KSDS01  )
  RIDFLD=  000                 F0F0F0

 RECORD AREA EXCEPTION @ OFFSET 00000017

           -....5....0....5...+
 CICS VT   .ORIGINAL.DATA.FOR.K 40D6D9C9C7C9D5C1D340C4C1E3C140C6D6D940D2
 VSAM      .ORIGINAL.BAD.DATA.F 40D6D9C9C7C9D5C1D340C2C1C440C4C1E3C140C6
```

Figure 74. DMF error reporting - different data returned

In this example, DMF has detected a difference between the VSAM and DB2 data at offset 0000017. This is a hexadecimal offset. Convert the hexadecimal offset to decimal, then add 1 to get the position of the field in the data set mapping information. DMF dumps 10 bytes of data before and after the offset where the difference is detected.

The DMF also records information when it detects a difference in CICS RESP codes.

```
*********************************************************************
* CICS VT COMPARE EXCEPTION                                         *
*********************************************************************

TRAN: K1P2  DIM: KSDS01  TERMINAL: 0019  TIME:15.31.3300  REQ:0000009

 EXEC CICS READ
  FILE(KSDS01  )
  RIDFLD=  YYYYYY               E8E8E8E8E8E8
  RBA
  EQUAL
  UPDATE

 RETURN CODE EXCEPTION

           RESP     RESP2    RCODE
 CICS VT   0000000D 00000050 810000000000
 VSAM      00000000 00000000 000000000000
```

Figure 75. DMF error reporting - different RESP codes

The RESP and RESP2 codes are hexadecimal values. In this case, CICS VT sets a RESP code of x'0D' indicating a NOTFND condition. The RESP2 code for NOTFND is always x'50'. The RCODE is not maintained by CICS VT and is produced for information purposes.

Additional DST parameters for DMF

There are two parameters in the DST that specifically related to DMF. They control how CICS VT should handle differences detected by DMF.

These parameters are specified at the DIM level as follows:

ERRORACT=ABEND|REPORT
Defines the action taken when an error is detected. The default is ABEND. When the default value is specified, a transaction abend code of 3139 is returned for data differences and a transaction abend code of 3141 is returned for a RESP code difference.

DUMP=NO|YES
Specifies whether CICS VT generates a transaction dump each time a difference is detected for the DIM. The default is NO. This parameter is only valid when the value of the **ERRORACT** parameter is ABEND.

DMF for batch programs

DMF is enabled at the VSAM data set level for batch programs. It requires an additional DD statement and an additional parameter in the CICS VT subsystem parameter list.

This is shown in the JCL for the CICS VT COBOL IVP batch job.

```
//VIDIVPCO JOB CICSVT,CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//RUNVID  EXEC PGM=VIDIVPCO,REGION=2M
//STEPLIB  DD  DISP=SHR,DSN=VID.SVIDLODE
//         DD  DISP=SHR,DSN=DB2.SDSNLOAD
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//IVPREPT  DD  SYSOUT=*,LRECL=130,RECFM=FBA
//SYSUDUMP DD  SYSOUT=*
//VIDKSDS@ DD  DSN=VID.VIDKSDS.DUMMY,DISP=SHR
//VIDKSDSV DD  DSN=VID.VIDKSDS,DISP=SHR
//VIDKSDS  DD  SUBSYS=(VIDS,DB2_ssid,VIDKSDS,VIDKSDSV)
//VIDTRCE  DD  SYSOUT=*,RECFM=FBA,LRECL=133,BLKSIZE=13300
//VIDTRCEP DD  *
TRACE P01,P08,DIM=VIDKSDS
```

Figure 76. Sample JCL with DMF enabled

The DD statement VIDKSDSV specifies the original VSAM data set. The DD statement name is also the fourth parameter in the SUBSYS statement. You can specify any unique name for the additional DD statement.

If you are using an alternate index PATH, add a new data set for the path entry and add the fourth parameter to the path SUBSYS statement.

Analyzing data differences

The most common errors reported by DMF are data differences.

Data differences can occur for a variety of reasons, such as any of the following situations:

- An error in the file mapping.
- An error in the data migration process.
- An error or bug in an IRD or FBE

When DMF detects a difference in the data retrieved from VSAM and DB2, diagnostic information is written to the job log. An example of the diagnostics follows:

```
VIDSS227 DUALMODE ERROR FOR SUBSYS=VIDZ VSAM DDNAME=ENTITLV
         RECORD DATA WAS NOT THE SAME AT OFFSET X"0036"
VIDSS219 DUMP OF DB2 RECORD:
VIDSS220 +0016 F0F0F0F0 F0F0F0F4 F1E4F4F1 E5F4F100 *000000041U41V41.*
VIDSS220 +0026 00004040 40404040 40404040 40404040 *..              *
VIDSS220 +0036 40404040 40404040 40404040 40404040 *                *
VIDSS220 +0046 40404040 40404040 40404040 40404040 *                *
VIDSS219 DUMP OF VSAM RECORD:
VIDSS220 +0016 F0F0F0F0 F0F0F0F4 F1E4F4F1 E5F4F100 *000000041U41V41.*
VIDSS220 +0026 00004040 40404040 40404040 40404040 *..              *
VIDSS220 +0036 C9C9C9D5 D5D5C6C6 C6404040 40404040 *IIINNNFFF       *
VIDSS220 +0046 40404040 40404040 40404040 40404040 *                *
```

Figure 77. DMF diagnostics for the data differences in batch programs

The diagnostics show that a difference in the data was detected at hexadecimal offset x'0036': the value in the record retrieved from VSAM is x'C9C9C9D5D5D5D5C6C6C6', and the value in the record built by CICS VT from DB2 is spaces.

Convert the hexadecimal offset X'0036' to decimal, then add 1 to get the position of the field in the data set mapping. In the example in Figure 77 on page 115, the field position is 55. Review the field mapping at position 55, and if the field is build by an FBE, review the FBE code.

DMF dumps 32 bytes of data before and after the error, shown as message number VIDSS220. The application program is abnormally terminated with a U3139 abend code.

Analyzing return code or reason code differences

A difference in the return code or reason code is most probably an indication that the data in the original VSAM data set is not the same as the data in the migrated data set. For example, a record exists in VSAM but not in DB2.

If the return code or reason code is different, CICS VT produces the following diagnostics:

```
VIDSS228 DUALMODE ERROR FOR SUBSYS=VIDZ VSAM DDNAME=ENTITLV
         FEEDBACK DIFFERENT ON PUT
         DB2  RPLRTNCD=00 RPLERRCD=00
         VSAM RPLRTNCD=08 RPLERRCD=08
         ACB MACRF=(ADR,SEQ,DIR,IN)
         RPL OPTCD=(SEQ,KEY) KEYL=0025
```

Figure 78. DMF diagnostics for return code differences in batch programs

The application program is abnormally terminated with a U3141 abend code.

Review the diagnostics that are written to the joblog. The significant fields are RPLRTNCD and RPLERRCD. Compare the VSAM and DB2 data by running the VIDREAD utility with DMF enabled. See “Step 4 - Verify data migration” on page 87 for sample JCL. If there is any difference between the data in VSAM and DB2, the VIDREAD utility will abnormally terminate with the abend code U3139.

If there is an FBE on any field that is part of the key, review the FBE. If the file is being accessed by an alternate index and there is an FBE on any field that is part of the alternate index, review the FBE.

Operational considerations for DMF

When CICS VT detects a difference, the application program abends. Ensure that the data in DB2 and in VSAM is synchronized. Otherwise, DMF might report errors that are due to data differences as opposed to potential problems with CICS VT or user exits.

If you are using DMF with a batch program that updates data, any abend is likely to result in unsynchronized data. This is because DB2 forces updates to the migrated data set to be rolled back, whereas the original VSAM data remains partially updated. It is essential that you are aware of this and take the appropriate steps to resynchronize the data following an abend. This could mean that you need to take a backup of the original VSAM data set prior to running an update program with DMF enabled.

If the program is read-only, data recovery following an abnormal termination is not required.

DMF in CICS and batch opens the original VSAM data set. This means that you will encounter sharing issues if you try to use DMF in CICS and batch concurrently. If you make a copy of the original VSAM, you can concurrently test read-only programs. Programs that perform updates cannot be tested concurrently.

Chapter 7. Production cutover

The mapping process is performed once for each VSAM data set, normally in the development or test system where the bulk of testing takes place.

To move to a different system, copy the drivers and DBRMs. You should not copy the mapping data. See "Configuring CICS VT on additional systems" on page 24 for more information.

This chapter describes how you move CICS VT application components to a different system. It is primarily aimed at helping you move to a production environment, although most of it applies when you move to a different test environment such as quality assurance or system test. The term *target system* is used to apply to any system where you will use CICS VT.

Process overview

The CICS VT DIM and DDM driver modules are generated when the mapping of a data set is complete. The driver modules and the DDM DBRM need to be copied to the target system. The mapping data in the CICS VT system tables is only used again if you change mapping, after which new driver modules should be generated.

The DDM driver includes the SQL used to access the DB2 data. This SQL is dependent on the attributes of the DB2 table containing the migrated VSAM data. This means that the DB2 table must be identical in each target system. The CREATOR can vary, but the table name, column names, and attributes must be the same. The DBRM produced during DDM generation must be copied to the target system and bound against the DB2 objects that will contain the VSAM file data.

The key objectives of the cutover are to ensure that the period when the files are unavailable is kept to a minimum, and that there is no adverse effect on service levels. To achieve these objectives, the cutover process is split into three phases:

1. Pre-cutover tasks
2. Cutover tasks
3. Post-cutover tasks

If you are unable to complete the cutover successfully, or you encounter severe post-cutover problems, you might decide to fall back to VSAM. A method to fallback to VSAM is described in "Fall back to VSAM" on page 122.

Pre-cutover tasks

There are a number of tasks that can be performed in advance of the cutover. None of these tasks adversely affect the target system. There is a degree of interdependency between some of these tasks, so the following set of steps is the recommended sequence.

Step 1 - Create DB2 objects

Run the DDL that you used to create the objects in the DB2 test system. Some modifications to the DDL might be required, such as storage group name and

database name. You might also have to increase space allocations if the data set you worked with in the test system contained fewer records than the production system data set.

Step 2 - bind DDM driver package and plan

Copy the DIM and DDM driver modules and the DDM DBRM to the target system and bind the DDM package and the driver plan. You should specify EXPLAIN(YES) in the package bind to generate access path information. Review this information to ensure that the access paths are correct. A tablespace scan might indicate that a required index has not been created.

Step 3 - Create CICS resource definitions for DIM, DDM, and CICS VT exits

Create PROGRAM resource definitions for the DIM and DDM programs, as well as any CICS VT user exits, in CICS. Note that the DIM, DDM, and CICS VT exits should be defined as RELOAD(NO). Follow the recommendations described in "Defining CICS VT driver modules in CICS" on page 106.

Step 4 - Add the DB2ENTRY entries to CICS

Transactions that previously accessed the data as a VSAM data set will now access DB2. Each transaction that accesses DB2 should be defined in the CICS CSD, according to your site standards.

Step 5 - Copy data migration jobs

The jobs that you created to perform the initial data migration on the test system need to be run on the target system. You might have to adjust the SPACE parameters in the JCL if the file on the target system is a different size than in the initial test system.

Step 6 - Create dummy VSAM cluster

Create a dummy VSAM data set for each migrated file. This is described in detail in "Additional DD statement for dummy VSAM data set" on page 98. Depending on the nature of the target system, you might want to gather some performance data from your DB2 system before cutover. After cutover, gather the same data and compare results. This will help you to tune your DB2 system to ensure that performance levels are not degraded. Pay particular attention to EDM and buffer pool statistics.

Step 7 - Prepare batch JCL

You can make the necessary batch JCL changes prior to cutover to the target system. Remember to add the DIM name to the migrated file exceptions module VIDFMSPX.

Cutover tasks

The file data is unavailable during the production cutover, so it is essential that you perform the minimum number of tasks during the cutover. It is also essential that you run these tasks successfully on another system first. The following sequence is recommended.

Step 1 - Disable access in CICS

The first step is to disable access to the file in CICS. This can be achieved by stopping the CICS region, or stopping the file using the CEMT transaction. You should stop alternate index data sets as well as base cluster data sets.

Step 2 - Back up the VSAM file

You should back up the VSAM file. This provides you with a fast fallback capability if you do not complete the cutover and decide to fall back to VSAM.

Step 3 - Unload the VSAM file

Run the CICS VT VIDUNLOD program to unload the VSAM file to a sequential data set.

Step 4 - Convert the data

Run the CICS VT VIDLOAD program to convert the unloaded file into the DB2 LOAD utility format.

Step 5 - Load the data into DB2

Use the DB2 LOAD utility to load the converted data into DB2. If you specify LOG NO, you should take a backup of the tablespace afterwards.

Step 6 - Test the migration

You should verify that the migration has completed successfully. You can do this by creating a version of the VIDUNLOD job with the necessary JCL changes to enable access with CICS VT.

Step 7 - Define file in the CICS VT DST

You need to add an entry for the file in the CICS VT data set control table. See "Adding a new data set to the CICS VT data set table" on page 104 for details on how to do this.

Step 8 - Restart the CICS-CICS VT interface

To use the new CICS VT DST, issue the VTMI RESTART WARM or COLD transaction.

Step 9 - Enable access to the migrated data set

Use CEMT to enable the file, as shown in the command that follows:

```
CEMT SET FIL(filename) ENA
```

Step 10 - Remove DIM from VIDFMSPX

Update VIDFMSPX to remove the DIM. Assemble a new VIDFMSPX.

Post-cutover tasks

The main objective is to ensure that there are sufficient resources available to support the change in the DB2 workload, and that the CICS-DB2 RDO interface is able to handle the increased CICS-DB2 activity. A number of areas must be monitored to ensure that there are no problems.

DB2 performance monitoring

The initial task is to ensure that the hit rates for the DB2 buffer pools and EDM pools are maintained. Gather performance data and compare it with the results taken prior to cutover. The DB2 DSNZPARM parameter CTHREAD limits the total number of DB2-CICS threads. You might have to increase this to handle the additional workload. See the *CICS DB2 Guide* for more information about tuning.

CICS-DB2 interface monitoring

The DB2 utility DSNC DISPLAY STATISTICS provides usage information for the DB2-CICS interface. Use this information to identify possible queuing or waiting for resources. The main parameters that might need to be increased to handle the additional CICS-DB2 workload are **THRDLIMIT** and **TCBLIMIT**.

DB2 page set monitoring

You will have to reorganize the new DB2 index and tablespace page sets based on the rate and nature of update activity. Implement a monitoring strategy to ensure that there is no impact on access times due to disorganization.

Fall back to VSAM

If you decide that you have to convert a migrated data set from DB2 back to VSAM, uses a 4-step procedure:

Step 1 - Unload the data from DB2

Run the VIDUNLOD program against the migrated data set. Because the data set has been migrated to DB2, enable the JCL to run under CICS VT. An example is shown in Figure 79.

```
//VIDUNLOD EXEC PGM=VIDUNLOD
//STEPLIB   DD DSN=appl.DRIVERS.LOAD,DISP=SHR
//          DD DSN=VID.SVIDLODE,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//FILEIN@   DD DSN=dummy.VSAM.dataset,DISP=SHR
//FILEIN    DD SUBSYS=(ssi,DB2id,dimname)
//FILEOUT   DD DSN=vsam.file,DISP=SHR
```

Figure 79. Sample JCL to unload a file from DB2

The output data set FILEOUT will have an 8-byte prefix for KSDS files and a 12-byte prefix for RRDS files. In both cases, the first four bytes will be the file type (KSDS, RRDS, KSDV, or RRDV). The next four bytes will contain the length of the record, in binary format. For RRDS data sets, the next four bytes contain the relative record number. The remainder of the record will be the data.

Step 2 - Define the VSAM clusters

Use the IDCAMS utility to delete and define the base VSAM cluster.

Step 3 - Load the data to VSAM

Use the CICS VT batch utility program VIDLOADV to load the data from DB2 into a VSAM data set. Sample JCL is provided in the member VIDLOADV in VID.SVIDSAMP and is also shown in next:

```
//VIDLOADV  EXEC PGM=VIDLOADV
//STEPLIB   DD DSN=VID.SVIDLODE,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//LOADIN    DD DSN=vsam.DB2unload,DISP=SHR
//LOADOUT   DD DSN=vsam.file,DISP=SHR
```

Figure 80. Sample JCL to load VSAM file

The steps in this fallback process can be combined into a single job to minimize the time it takes to fall back.

Step 4 - Define the alternate index and path cluster

Use the IDCAMS utility to define the alternate index cluster and path and build the alternate index path entries.

Step 5 - Remove the DIM from CICS and batch access

If you are falling back to VSAM temporarily, you can use the VTMM transaction to set the CICS file migration status to INACT. Otherwise, remove the entry for this DIM from the CICS VT DST.

For temporary batch fallback to VSAM, add the DIM to VIDFMSPX. Otherwise, reverse the JCL changes.

Chapter 8. CICS VT operational and recovery considerations

When a VSAM record is updated by an application program, the change to the data is made immediately. When you update a DB2 row, the change is only made after a COMMIT, a CICS SYNCPOINT or TERM. When you convert a data set to DB2 using CICS VT, you must consider how this difference between VSAM and DB2 affects your existing operational and recovery procedures.

You must also consider how to integrate the DB2 elements of CICS VT migrated data sets into your existing DB2 environment. For example, what DB2 collection will you use for your CICS VT driver packages? How does the migration affect your backup and recovery procedures?

You must understand these implications on your operational and recovery procedures before migrating any production data sets from VSAM to DB2.

Implications for CICS programs

CICS programs that process CICS VT migrated data sets access DB2. If you do not already have CICS programs that issue SQL calls, read the *CICS DB2 Guide* before enabling CICS VT in your CICS system.

DB2 administration

For each migrated VSAM data set and alternate index path, a CICS VT DDM driver module is used to access DB2. Bind the DB2 package for the DDM into the DB2 plan that the CICS transaction uses. The default collection for DDM packages is called VIDCOLL. If your existing CICS DB2 programs use a single plan, add the collection VIDCOLL to the package list for this plan.

If your CICS transactions use different plans, bind the collection VIDCOLL into each one.

Data recovery

Changes to DB2 tables are accumulated until one of the following events occur, at which point the updates are committed:

- A CICS SYNCPOINT call is processed.
- A CICS EXEC DLI TERM call is processed.
- The application program ends normally.

Updates to VSAM files are handled in the same way by CICS.

Changes to DB2 tables are backed out in the following circumstances:

- A CICS ABEND call is issued.
- The application program abends.
- The application program issues a ROLLBACK TO SAVEPOINT call.

Updates to VSAM data sets are backed out in the first two cases. If you have existing CICS programs updating DB2 and VSAM resources in the same unit of work, a ROLLBACK TO SAVEPOINT call issued by your application program will

undo changes to CICS VT migrated DB2 tables. You should consider how this affects your CICS application programs before migrating your production data.

When you migrate VSAM data sets to DB2 using CICS VT there are no other specific application recovery implications for CICS programs.

Implications for batch programs

Batch programs handle updates to DB2 and VSAM resources differently to CICS application programs. For example, there is no logging of VSAM updates in a batch program.

To deal with this, there are two different processing methods for batch programs in CICS VT. The default method processes all SQL calls in the same DB2 thread. The second method, which is enabled at an individual batch program level, uses MVS subtasking to allow you to have multiple, independent threads to DB2.

CICS VT subtasking incurs a performance overhead and should only be used when necessary. The next sections explain CICS VT subtasking and define when it should be used.

CICS VT subtasking

Batch programs that do not contain SQL do not require subtasking. Batch programs that do contain SQL might require subtasking, based on the following factors:

- If either VSAM or DB2 resources are updated.
- How the connection to DB2 is established.

The following example demonstrates when to use CICS VT subtasking and what the results are.

Assume that you have an application program that updates two VSAM data sets called FILEA and FILEB. These data sets have been migrated to DB2 tables called VID_TABA and VID_TABB using CICS VT. The application program also contains SQL calls and updates a DB2 table called SQL_TAB. The program issues DB2 commit calls at certain intervals. The effect of these commit calls depends on whether CICS VT subtasking is being used or not.

- If CICS VT subtasking is not enabled, updates to tables SQL_TAB, VID_TABA, and VID_TABB are committed by each explicit commit issued by the application program.
- If CICS VT subtasking is enabled, only the updates to SQL_TAB are committed by explicit commit calls issued by the application program. Updates to VID_TABA and VID_TABB are committed or rolled back at the end of the program.

Situations where subtasking is optional

CICS VT uses the DB2 call attachment facility (CAF) to connect to DB2. For application programs that access VSAM data sets that have been migrated using CICS VT and issue native SQL calls using the DB2 CAF attachment, there are two scenarios to consider:

- Scenario 1: If the application program issues an SQL call before accessing a CICS VT migrated data set, CICS VT subtasking is not required. CICS VT will use the DB2 thread created by the application program CONNECT and OPEN to process SQL calls to access the migrated VSAM data sets.

- Scenario 2: If the application program issues an SQL call after accessing a migrated data set, the application program DB2 CONNECT request receives a return code of 200 and a reason code of X'00C10201' if subtasking is not enabled. If your application program tests this return code, it might signal an error and abend the program. A DB2 return code of 0000 is returned if subtasking is enabled.

In both of these scenarios, subtasking should be used if your application program issues explicit SQL commit calls.

Situations where subtasking is mandatory

If your batch programs use the DB2 TSO attachment, you must use subtasking. Otherwise, the following abend is issued when CICS VT tries to establish a CAF connection:

```
VID2CAF  db2ssid CONNECT ERROR HAS OCCURRED, PLAN=dimname
RETURN CODE = 0008  REASON CODE = 00F30049
```

This abend is issued because of the conflict between the application program running under the TSO DB2 attach and CICS VT trying to establish a connection to DB2 using CAF.

As an alternative to using subtasking, you might consider converting these programs to use the CAF.

Enabling subtasking

You enable CICS VT subtasking at an individual batch program level, by adding the following DD statement to your batch job JCL:

```
//VIDSUBSW   DD  DUMMY
```

You can disable subtasking by removing the statement.

CICS VT multiple TCB support

CICS VT supports batch programs that create multiple TCBs to access VSAM files. Support is enabled by adding the following DD statement to your batch job JCL:

```
//VIDTCBSW   DD  DUMMY
```

Note that multiple TCB support and CICS VT subtasking are mutually exclusive. If you try to enable both at the same time, the batch job terminates abnormally.

DB2 connection exit

By default, CICS VT uses the DB2 call attach facility. An exit is provided to enable a different connection to be used.

You can override the default DB2 CAF connection by providing a user exit. The exit establishes a DB2 connection and returns control to CICS VT.

The exit name must be VIDCATTE and is coded in assembler. A sample exit is provided in *my*.SVIDSAMP.*custom* member VID1CATT. The exit uses a parameter list defined by the macro VIDCATTP, which is provided in *my*.SVIDSAMP.*custom* member VID1CATP.

The exit is called once for each job step, prior to connecting to DB2. The parameter list provided by CICS VT to the exit contains the addresses of the following parameters:

- The DB2 subsystem from the SUBSYS JCL statement

- The DB2 plan name from the SUBSYS JCL statement, or the VIDPLAN DD statement if supplied
- A save area for the exit to use

The exit uses a return code in register 15 to direct CICS VT as follows:

- If RC=0, a DB2 connection has been established by the exit
- If RC=4, no DB2 connection has been established and CICS VT should establish a CAF connection.

On return, the relevant exit parameter list fields are as follows:

- If RC=0, the SQL processing interface address CATTSQLP *must* be provided
- If RC=4, optionally the DB2 subsystem ID that the batch job should use
- If RC=4, optionally the plan name that the batch job should use. Note that this overrides the VIDPLAN DD statement if supplied

If CICS VT subtasking is active, the exit is not called.

DB2 administration

The DB2 packages for the CICS VT DDMs must be available to the application program at run time. If you are not using subtasking, CICS VT uses the DB2 plan for the first DDM that is used by the application program. Therefore, you must ensure that all DDM packages are bound into all of the DDM plans. The most effective way to do this is to use the same collection for each package and bind the entire collection into every DDM plan.

If you are using subtasking, CICS VT uses a different plan for each migrated VSAM data set.

Overriding the default DB2 plan name

In some circumstances you might want to run a batch job using bind parameters that are different to the default plan bind parameters. For example, you could have a read only program and you want to take advantage of DB2 lock avoidance using the `CURRENTDATA(NO)` parameter, or you could have programs that can handle the processing of uncommitted updates.

The third parameter in the subsys JCL statement contains the name of the DIM, which is also the default DB2 plan name. To override the default plan name, add the following DD statement to your JCL:

```
//VIDPLAN DD *
plan-name
```

When CICS VT creates a DB2 thread, it uses the plan *plan-name*.

To illustrate this, assume that you have a batch job that accesses three migrated VSAM data sets with DIM names VSAM1, VSAM2, and VSAM3. VSAM1 and VSAM2 are accessed in read-only mode, and VSAM3 is potentially updated. You want to access the data sets as follows:

- VSAM1 - This file can be accessed without locking and can process uncommitted updates.
- VSAM2 - This file can be accessed without locking and must only read committed updates.

- VSAM3 - This file is updated and must be accessed using regular page or row locking.

To use the facility to override the default DB2 plan name requires the following actions:

1. Choose a unique DB2 plan name for this batch job. For example, this could be the MVS job name or the application program name.
2. Bind the DDM DBRM for VSAM1 with attributes of `CURRENTDATA(NO)` and `ISOLATION(UR)`. Bind this package into a collection that is reserved for VT DDM packages with these attributes, such as VIDURNO.
3. Bind the DDM DBRM for VSAM2 with attributes of `CURRENTDATA(NO)` `ISOLATION(CS)`. Once again a unique collection name is required, for example VIDCSNO.
4. Use the following command to bind the packages from steps 2 and 3 and the default package for VSAM3 into the plan that will be used for this batch job.

```
BIND PLAN(JOBNAMEP)         +
     PKLIST(VIDURNO.VSAM1,  +
            VIDCSNO.VSAM2,  +
            VIDCOLL.VSAM3)
```

5. Add the following DD statement to the batch job:

```
//VIDPLAN  DD *
JOBNAMEP
```

You can use the DB2 plan override capability for jobs that run in either non-subtasking or subtasking modes. You can also override the default plan name using the VIDCATTE connection exit.

If you specify a plan name that does not exist or a null input, the batch job abends on the first file open in your application program. The abend code is S013-C8.

Data recovery

After a data set has been migrated by CICS VT, commit and backout processing is managed by DB2. Updates are committed if the program ends normally, and backed out if the program abends. This affects your operational and recovery procedures and you must take steps to ensure that you manage the DB2 data securely and effectively.

- Convert your backups to DB2 image copies. You can either remove or change backups taken before running update jobs to the DB2 quiesce utility.
- Disable any recovery jobs that are automatically initiated by job scheduling systems.

You should also consider how CICS VT affects your disaster recovery procedures.

Table level locking

There might be occasions when you want a batch job to have exclusive access to a migrated data set. This is equivalent to specifying `DISP=OLD` in your JCL. CICS VT provides a facility to exclusively access a migrated data set by issuing the **`LOCK`** *tabx* **`IN EXCLUSIVE MODE`** command when your application program opens the data set.

To use the facility, add the following DD statement to the JCL for your batch job:

```
//VIDLOCK  DD *
dim-name1
dim-name2
dim-name3
```

For each DIM name specified, CICS VT locks the table in exclusive mode. The facility is available for batch jobs that run in either non-subtasking or subtasking modes.

Table level locking and package bind implications

Using the table lock facility imposes certain rules as follows:

- You must bind the DDM DBRM into the DB2 collection VIDLOCK.
- You must specify `RELEASE(DEALLOCATE)` in the package bind.
- The VIDLOCK package must be the first package in the package list in the plan bind, and `VIDCOLL.*` must be included as the last package in the package list.
- You must add the VIDPLAN DD statement and specify the name of the plan.

If one or more of these rules are not met, CICS VT issues a U3127 abend during open processing, and the program is abnormally terminated.

These rules are required in case you add SQL commit statements to your program, in which case the table lock would otherwise be released by a commit.

Table level locking and IRD implications

There are implications if you use the table lock facility for a DIM that has an IRD exit. CICS VT assumes that the IRD exit is being used to split a data set into multiple DB2 tables. To achieve the equivalent of DISP=OLD, a table accessed by an IRD exit must also be locked. You must take this into consideration when you write an IRD exit, and add the appropriate code.

During an open call for a DIM that is specified in the VIDLOCK DD statement, CICS VT calls the IRD exit. In the parameter list that CICS VT provides to the exit, the field IRDFUNCT has a value of `'X'`. Before the exit ends and returns control to CICS VT, the field IRDRET must be set to a value of `'X'`. When CICS VT resumes control, the value of IRDRET is checked. If it is not `'X'`, CICS VT issues a U3126 abend and the program is abnormally terminated

It is not essential for your IRD exit to issue the LOCK TABLE SQL statement, but IRDRET must be set correctly to avoid a U3126 abend.

Adding SQL commit calls

Existing application programs that perform many updates may cause DB2 locking issues.

The only way to reduce the number of concurrently locked pages in DB2 is to issue SQL COMMIT calls. If you encounter locking issues, you may have to consider changing your application programs.

The least disruptive way to do this is to code a new program that simply issues a COMMIT. Code this program to uses either the DB2 remote recovery services attachment (RRSCAF) or the call attach facility (CAF). Instructions for both methods are in the appropriate DB2 manual. Add code to the long-running update programs to call the new module.

Bind the DBRM from the commit program into the same collection as the CICS VT DDM driver.

Be aware of the following considerations:

- Your new program does not need to include code to connect to DB2. Providing that you call it after a number of VSAM calls to a migrated data set have occurred, a DB2 thread will already exist.
- You must carefully consider how to establish the optimum points in your program to call the commit module.
- You must consider very carefully the implications of restarting your application program if an abend occurs after one or more commit calls have been processed.

Using this technique is only appropriate if your long running batch programs only update VSAM files. It is not appropriate if other resources such as DB2 or MQSeries are involved.

CICS VT and DFSORT

You can use DFSORT to sort or unload a VSAM data set.

CICS VT supports DFSORT programs, but you must specify additional JCL parameters in the DD statement that defines the VSAM data set you have migrated. The additional parameters are shown in the following JCL statement:

```
//SORTIN   DD SUBSYS=(ssid,DB2id,dimname),
//         RECFM=ff,LRECL=llll,BLKSIZE=bbbb
```

The value of *ff* should be either F or V. *llll* is the maximum VSAM record length value. If the VSAM records are fixed length, BLKSIZE *bbbb* is also the maximum VSAM record length value. If the VSAM records are varying length, the BLKSIZE value *bbbb* is *llll* + 4.

CICS VT and IDCAMS

CICS VT supports a subset of IDCAMS PRINT and REPRO functions. There is no support for other IDCAMS functions.

PRINT

You can use PRINT with a CICS VT migrated data set. The following PRINT options are supported:

- SKIP(*n*)
- COUNT(*n*)
- Any combination of SKIP and COUNT

REPRO

Using a CICS VT migrated file as `INFILE`, the following REPRO options are supported:

- SKIP(*n*)
- COUNT(*n*)
- Any combination of SKIP and COUNT

Using a CICS VT migrated file as `OUTFILE` requires an additional DD statement if either `REPLACE` or `REUSE` is specified. In addition, the LRECL parameter must be added to the output DD statement. Figure 81 on page 132 shows sample JCL for IDCAMS with REPRO REPLACE.

```
//REPLT    EXEC PGM=IDCAMS,REGION=0M
//STEPLIB   DD DISP=SHR,DSN=CICSVT.TEST.DRIVERS
//          DD DISP=SHR,DSN=CICSVT.TEST.EXIT.LOAD
//          DD DISP=SHR,DSN=VID.VIDLOAD
//          DD DSN=SYS1.LINKLIB,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//IDCIN     DD DISP=SHR,DSN=CICSVT.TEST.DIMREPL.INPUT
//IDCOUT@   DD DISP=SHR,DSN=CICSVT.TEST.DIMREPL.DUMMY
//IDCOUT    DD DISP=SHR,SUBSYS=(VIDZ,DB2B,DIMREPL),LRECL=llll
//
//SYSIN     DD *
  REPRO INFILE(IDCIN) OUTFILE(IDCOUT) REPLACE SKIP(2) COUNT(4)
//VIDIDCIN  DD *
DIMREPL REPLACE
```

Figure 81. Sample REPRO JCL

The additional DD statement required for REPRO `REPLACE` or `REUSE` is VIDIDCIN. The format of the input is:

dim-name `REPLACE` or

dim-name `REUSE`

dim-name must start in position 1, and `REPLACE` or `REUSE` must start in position 9.

The LRECL value *llll* must be the length of the longest record in the input file + 4.

Note: Use *dim-name* `REUSE` with caution. CICS VT has no access to the IDCAMS SYSIN statements and when `REUSE` is specified in VIDIDCIN, every row in the appropriate DB2 table is automatically deleted for the specified DIM.

Multiple VIDIDCIN statements

You can specify multiple VIDIDCIN input statements, but you can only specify a *dim-name* once. For example, assume that you are using IDCAMS to copy two input sequential data sets to the same output VSAM file as follows:

```
REPRO INFILE(IDCIN1) OUTFILE(IDCOUT) REUSE
REPRO INFILE(IDCIN2) OUTFILE(IDCOUT) REPLACE
```

Without CICS VT, the VSAM file contains only those records from IDCIN1 after the first REPRO statement is processed. After the second REPRO statement is processed, the VSAM file contains a combination of updated and new records from IDCIN2. With CICS VT, you must split this into separate job steps. The first job step will have the first REPRO statement and the VIDIDCIN statement will be:

dim-name `REUSE`

VIDIDCIN for the second job step will be:

dim-name `REPLACE`

Multiple REPRO statements in the same job step but for different VSAM files is supported.

General IDCAMS notes

- You must add the library containing the IDCAMS module to the batch job STEPLIB.

- Range processing using TO/FROMKEY, TO/FROMNUMBER, and TO/FROMADDRESS is *not* supported by CICS VT.
- You cannot use the CICS VT dual mode facility with IDCAMS.

Chapter 9. Maintaining CICS VT migrated applications

Periodically, you might need to change the application programs that access CICS VT migrated files. You might also decide to change some aspect of a DB2 table containing CICS VT migrated data. This chapter helps you understand how CICS VT affects application maintenance.

The only times you need to consider CICS VT during routine application maintenance are:

- When you are changing a call to a migrated data set.
- When you add an SQL call to a program for the first time.

This chapter looks at both of these situations and explains what you need to do for these application changes. It also identifies the circumstances when you need to regenerate CICS VT drivers.

Changing existing VSAM calls

Each CICS VT DDM driver module contains all of the SQL to simulate every possible VSAM call. This means that you can change any existing calls in an application program without affecting CICS VT. You can also add new VSAM calls to an existing program without affecting CICS VT.

As an alternative to changing an existing VSAM call, you could consider replacing it with an equivalent SQL call. This allows you to use the full functionality of SQL. You can also add a new VSAM call to an existing application program.

No CICS VT actions are required in these situations.

Adding a new VSAM call

No CICS VT actions are required when you add a new VSAM call to an application program. This applies to calls for VSAM data sets or CICS VT migrated data sets.

Adding SQL to an existing program

Instead of adding a new VSAM call, you should consider adding an SQL call. You might want to add functionality to a program, or replace or modifying an existing VSAM call with an SQL call. There are potential CICS VT issues to consider. These are detailed in Chapter 8, “CICS VT operational and recovery considerations,” on page 125.

There are no specific CICS VT issues to consider if you are adding SQL to a program that already contains SQL.

Regenerating CICS VT drivers

You need to regenerate CICS VT drivers if you change a CICS VT DB2 table. For example, if you add a new column or change the attributes of an existing column you must regenerate the drivers to let CICS VT take account of the changes. You should also ensure that DB2 is still choosing an efficient access path by adding the EXPLAIN(YES) parameter to the package bind and reviewing the rows in the PLAN_TABLE table.

For details on how to generate CICS VT drivers, see “Generating CICS VT runtime modules” on page 53.

Chapter 10. Tracing and diagnostics

This chapter outlines some of the debugging capabilities of CICS VT.

CICS VT tracing

The trace facility traces the flow of VSAM calls processed by CICS VT. User-specified input parameters define the data sets to be traced and the level of detail to be included in the trace output.

Up to ten data sets can be selected for tracing simultaneously. There are eight different CICS VT events that can be traced. These are referred to as trace points.

Trace points

The available trace points and the resulting trace data are as follows:

Trace Point	Trace Data
Point 1	Start of VSAM Call Data presented: Request number Call type and data set type DIM and DDM names Key value used (where appropriate) RPLNUM (where appropriate) DD name for file (where appropriate)
Point 3	Open a DB2 cursor Data presented: Name of DB2 table DB2 key value for the cursor
Point 4	DB2 Cursor Fetch Data presented: Name of DB2 table DB2 key value for cursor DB2 key value of fetched row
Point 5	Close DB2 Cursor Data presented: Name of DB2 table
Point 6	Build Record Field from DB2 Record Data presented: DB2 source column data used as input Data set destination field built

Trace Point	Trace Data
Point 7	Build DB2 Column from Record Field Data presented: VSAM field source data value Name of FBE (if user exit builds field) DB2 column value
Point 8	End of Call Data presented: Return code
Point 9	After a DB2 INSERT, REPLACE or DELETE call Data presented: Contents of the DB2 row SQL return code

Trace points 3, 4, and 5 trace SQL events and show the SQL code returned by DB2. Trace point 6 for GET type calls and point 7 for WRITE or REWRITE calls can produce many lines of output. Only use these when it is essential to see data at the field level.

Trace parameters

The trace parameters can be specified as SYSIN DD *, or in an 80-byte sequential data set or partitioned data set member. The parameter list has the following format:

```
TRACE P0n,P0n,DIM=dim-name-1,dim-name2,dim-name-3
```

TRACE is a required keyword that starts in column 1 and must be followed by a space. The parameter descriptions are as follows:

P0*n*,P0*n*
Specify the trace points to be activated. The values are P01 through to P09, except for P02 which is reserved. Separate each trace point with a comma.

DIM=*dim-name-1*,*dim-name-2*, *dim-name-3*
Specify the required DIMs for tracing. Separate each DIM with a comma. You can specify up to a maximum of 10 DIMs. Specify the a read-only DDM module name instead of the DIM name for jobs that use the read-only DDM. Note that the trace records still refer to the DIM name in these circumstances.

You can continue specifying tracing in a second statement if needed. The first statement must end with a comma and the list of DIMs must be continued on the second statement starting in column one as follows:

```
TRACE P01,P03,P04,P05,P08,P09,DIM=VIDKSDS,VIDKA01,VIDKX02,
VIDKX03,VIDKX04
```

Optionally, you can trace a subset of calls. This can be useful for tracing batch programs that issue many calls and where you are only interested in specific calls. To trace a subset of calls, add an additional trace input statement after the TRACE statement, as follows:

```
TRACSET CALLS=nnnn,SKIP=nnnnnn
```

TRACSET is a required keyword that starts in column 1 and must be followed by a space. The parameter descriptions are as follows:

CALLS=*nnnn*
Specify four digits defining the number of calls to be traced. Pad with leading zeros.

SKIP=*nnnnnn*
Specify one to 6 digits defining the number of calls to be skipped before tracing begins. Pad with leading zeros.

The CALLS and SKIP parameters can be used individually but if both are defined, CALLS must come first. Here are some examples:

Example 1

```
TRACE P01,P03,P04,P05,P08,DIM=VIDKSDS
TRACSET CALLS=0013,SKIP=000007
```

Example 2

```
TRACE P01,P03,P04,P05,P07,P08,P09,DIM=VIDKSDS,VIDKX01
TRACSET SKIP=001990
```

Activating a trace in batch jobs

Activate the trace for batch jobs by adding DD statements for the parameter data set and the trace output to the JCL. Below is an example of the DD statements and the parameters:

```
//VIDTRCE  DD  SYSOUT=*,RECFM=FBA,LRECL=133
//VIDTRCEP DD  *
TRACE P01,P02,P03,P04,P05,P06,P08,P09,P07,DIM=VIDKSDS
TRACSET CALLS=0035,SKIP=000360
```

The VIDTRCEP DD can be an 80-byte data set instead of SYSIN cards.

Activating the trace in CICS

To activate the trace facility in a CICS region, run the supplied transaction VTMT. The following screen is displayed:

```
--------------------------  CICS VT TRACE STATUS  ----------- this termid => 0037

    DIMS BEING TRACED:     DIM0001_ DIM0002_ DIM0012_ ________ ________
                           ________ ________ ________ ________ ________

    TRACE POINTS ACTIVE:   01   05   07   __   __   __   __   __   __

    DUMP POINTS ACTIVE:    __   __   __   __   __   __   __   __   __

    TERMIDS BEING TRACED:  0037 0051 ____ ____ ____
                           ____ ____ ____ ____ ____

              s ACTIVATE TRACE    _ DE-ACTIVATE TRACE

  Positions of the standard Trace Points :-
  01 - start of VSAM call                 06 - after Field Build Exit
  02 - Reserved                           07 - after DB2 Column built
  03 - DB2 Table Cursor Open              08 - VSAM return Code
  04 - DB2 Table Read                     09 - DB2 ISRT / DLET / REPL
  05 - DB2 Table Cursor Close

 MAKE CHANGES AND ENTER 'S' TO SELECT ACTION
 CLEAR:END SESSION
 TRACE IS ACTIVE
```

Figure 82. VTMT trace parameter screen

You must specify at least one DIM and one termid. Alternatively, you can specify *ALL as either the DIM or termid. Use termid *ALL to trace non-terminal related transactions.

Using *ALL for DIM and termid at the same time may result in significant trace output depending on your CICS workload, and may cause some degree of system degradation while the trace is active.

Activate and de-activate trace by selecting either ACTIVATE TRACE or DE-ACTIVATE TRACE options. The current trace status is displayed at the foot of the screen.

In most cases, trace points 01, 03, 04, 05, and 08 provide the most useful diagnostics. Avoid using trace points 06 and 07 unless you need to capture field-level information.

The output from the trace is sent to the CICS destination VIDT, which is defined during CICS VT configuration and setup. This destination relates to the DD statement VIDTRC in your CICS JCL, which can be either a 133-byte sequential data set or SYSOUT.

Internal diagnostics trace

CICS VT has an internal diagnostic trace facility, to be used only under the direction of your CICS VT software supplier. If you are asked to provide this trace, your software supplier will supply the input parameter format.

Identifying SQL errors

CICS VT translates SQL codes into appropriate VSAM return codes and reason codes, which are returned to your application program.

For example, the SQL code -803 indicates that an inserted or updated value violates a DB2 uniqueness constraint. In other words, an attempt has been made to

insert a duplicate key. CICS VT translates this condition into a matching condition in VSAM and returns the appropriate information to the application program.

It is almost inevitable that CICS VT will encounter certain conditions in DB2 that do not translate exactly into an equivalent VSAM return code and reason code. For example, if a DB2 tablespace is only available for utility processing, DB2 returns a -904 SQL code. The way that CICS VT reports these conditions varies between batch and CICS programs.

For batch programs, CICS VT writes a formatted SQL communication area (SQLCA) to the system console log and the batch job log. Here is an example:

```
VIDCAF    SUB SYSTEM = DB2B   PLAN NAME = VIDKSDS
VIDCAF                PRECOMPILER SEQUENCE NUMBER =  756
VIDCAF    DSNT408I SQLCODE = -904, ERROR:  UNSUCCESSFUL EXECUTION CAUSED BY
VIDCAF                                     AN UNAVAILABLE RESOURCE. REASON CODE
VIDCAF                                     00C90086, TYPE OF RESOURCE 00000200,
VIDCAF                                     AND RESOURCE NAME VIDTSTDB.VIDKSDS
VIDCAF    DSNT418I SQLSTATE = 57011 SQLSTATE RETURN CODE
VIDCAF    DSNT415I SQLERRP  = DSNXRRC SQL PROCEDURE DETECTING ERROR
VIDCAF    DSNT416I SQLERRD  = 111  13172746  0  13223106  -974970876  12714050
VIDCAF                        SQL DIAGNOSTIC INFORMATION
VIDCAF    DSNT416I SQLERRD  = X'0000006F'  X'00C9000A'  X'00000000'  X'00C9C4C2'
VIDCAF                        X'C5E32004'  X'00C20042'
VIDCAF    SQL DIAGNOSTIC INFORMATION
```

Figure 83. CICS VT SQL error reporting in batch

In CICS, the error information is written to the DD statement VIDDMP. This is an optional CICS VT DD statement that you should add to every CICS system or region that runs CICS VT. See "Modifying the CICS startup JCL" on page 14 for more details. The content of a VIDDMP DD statement is shown in Figure 84:

```
*******************************************************************************
*                                                                             *
*             C I C S  V T  F O R M A T T E D   D U M P                       *
*                                                                             *
*        DATE: 14.11.2009                            TIME: 15.56.3322         *
*                                                                             *
*        TRANSACTION: VTMV     TERMINAL: 0010                                 *
*                                                                             *
*        CALLTYPE:    GET      DIMNAME : VIDKSDS                              *
*                                                                             *
*******************************************************************************

DSNT408I SQLCODE = -904, ERROR:  UNSUCCESSFUL EXECUTION CAUSED BY AN
         UNAVAILABLE RESOURCE. REASON 00C90081, TYPE OF RESOURCE 00000200, AND
         RESOURCE NAME VIDTSTDB.VIDTSA
DSNT418I SQLSTATE   = 57011 SQLSTATE RETURN CODE
DSNT415I SQLERRP    = DSNXRRC SQL PROCEDURE DETECTING ERROR
DSNT416I SQLERRD    = 111  13172746  0  13223106  -974970876  12714050 SQL
         DIAGNOSTIC INFORMATION
DSNT416I SQLERRD    = X'0000006F'  X'00C9000A'  X'00000000'  X'00C9C4C2'
         X'C5E32004'  X'00C20042' SQL DIAGNOSTIC INFORMATION
```

Figure 84. CICS VT SQL error reporting in CICS

For certain DB2 errors, diagnostic data is written to the DB2 master address space (ssidMSTR) by DB2.

Calling technical support

When reporting a CICS VT problem, you need to provide technical support with diagnostics information. Use the following table as a guide:

Type of Problem	Diagnostics Required
System abend, such as S0C1 or S0C4	Full SYSUDUMP or SYSABEND. You must ensure that any dump suppression from LE/370 or Abend-Aid for example is removed.
CICS transaction abend	A formatted CICS transaction dump and a CICS VT trace for the failing program. The MSGUSR DD data is optional but often useful.
Incorrect data returned	CICS VT trace, with a minimum of trace points 1, 3, 4, 5, and 8. See "CICS VT tracing" on page 137.

In some cases for CICS problems, you might need to supply screen prints from a CICS EDF session.

Chapter 11. Messages and Codes

CICS VT messages are split into a number of categories:

- User abend codes issued by batch programs. These are in the range 3100 to 3200.
- CICS abend codes in the range VIDA to VIDZ.
- Error messages issued during DIM driver generation. These have the prefix VIDGD.
- Error messages issued during DIM driver generation. These have the prefix VIDGP.
- Error messages issued during DB2 connection processing. These have the prefix VIDCA.
- Information messages issued during CICS VT tracing. These have the prefix VIDTR.
- Data conversion errors. These have the prefix VIDF.
- Error messages produced by the VIDMAPIN program. These have the prefix VIDIN.
- Information and error messages returned by the CICS VT batch subsystem. These have the prefix VIDSS.

CICS VT system abend codes

Abend Code	Abend Code Value
ABEND U(3100)	An error has occurred during VIDUNLOD processing. The associated job log message details the error.
ABEND U(3101)	An error has occurred during VIDLOAD processing. The associated job log message details the error.
ABEND U(3102)	CICS VT main task abend. Preceding ABEND information details the initial ABEND.
ABEND U(3103)	Module load failure in CICS VT initialization. The associated message names the module that could not be loaded.
ABEND U(3104)	Subtask initialization failed. Contact IBM Technical Support.
ABEND U(3105)	There are not enough threads available for the current application program. Generate the DDM driver again specifying a greater number of concurrent threads.
ABEND U(3106)	There is no KEY field mapped. This is a mapping error.
ABEND U(3107)	An FBE exit returned an error return code. Register 3 contains a pointer to the VIDFBEP control block used by the exit.
ABEND U(3108)	An IRD exit returned an error return code. Register 3 contains a pointer to the VIDIRDP control block used by the exit.

Abend Code	Abend Code Value
ABEND U(3109)	An ABEND has occurred in the CICS VT abend handling routine. The SDWA cannot be located. Contact IBM Technical Support.
ABEND U(3110)	VIDCAF cannot locate call parameters to enable connection to DB2 . Contact IBM Technical Support.
ABEND U(3111)	User error. A program using a read-only DDM has attempted to issue an update call.
ABEND U(3112)	GETMAIN error in VSAM call analysis routine. Contact IBM Technical Support.
ABEND U(3113)	CICS VT does not support asynchronous RPL requests.
ABEND U(3114)	VIDPATHM has encountered a non-zero SQL code.
ABEND U(3115)	An invalid parameter has been defined in the input to a conversion routine. See either message VIDFC001 or VIDFC002 as necessary.
ABEND U(3116)	The VIDCONV S0C7 SPIE routine has been driven by an S0C7 abend from a program other than VIDCONV. Remove VIDS0C7 DD card to find out where.
ABEND U(3117)	OPEN error for VIDS0C7 card.
ABEND U(3118)	An error has occurred checking the conversion of null fields.
ABEND U(3119)	An error has occurred in driver generation error processing. See message VIDGP004.
ABEND U(3121)	The mapping facility has encountered an error during DB2 connection processing. See the explanation for message VIDCA002.
ABEND U(3122)	The mapping facility has encountered an error during DB2 open processing. See the explanation for message VIDCA003.
ABEND U(3123)	An error occurred building a CICS VT message. This would normally happen if the message block module were corrupt. Contact IBM Technical Support.
ABEND U(3125)	A driver loaded by CICS VT Version was generated by a previous release and is not compatible. Regenerate the DIM and DDM drivers for this data set.
ABEND U(3126)	The VIDLOCK DD statement includes a DIM with an IRD exit, and the field IRDRET has not been set to 'X' by the exit.
ABEND U(3127)	The VIDLOCK DD statement is specified in the JCL, and the package and plan do not adhere to the DB2 bind rules.

Abend Code	Abend Code Value
ABEND U(3131)	An error has occurred in the initialisation of the PIPI environment. Program CEEPIPI is not available. In CICS, add program CEEPIPI to the CSD. In batch, add cee.SCEERUN to the STEPLIB or JOBLIB.
ABEND U(3132)	An error has occurred in the initialisation of the PIPI environment. An error has been returned from the init_sub_dp call. R15 points to the return code. Check the LE/370 return code in the appropriate LE/370 manual.
ABEND U(3133)	An error has occurred adding the exit program to the usable PIPI environment. R15 points to the return code from the add_entry call. Check the LE/370 return code in the appropriate LE/370 manual.
ABEND U(3134)	The PIPI init program table is full and no more exit programs may be added for use. Contact IBM support.
ABEND U(3135)	An error has occurred calling the exit program. R15 points to the return code from the call_sub call. Check the LE/370 return code in the appropriate LE/370 manual.
ABEND U(3139)	The dual mode facility has detected a difference in the record areas from VSAM and DB2. Review the additional diagnostic messages produced on the JES job log.
ABEND U(3140)	The dual mode facility has detected a difference in the record lengths from VSAM and DB2. Refer to the preceding VIDSS226 message in the JES job log.
ABEND U(3141)	The dual mode facility has detected a difference in either the return code or reason code from VSAM and DB2. Review the additional diagnostic messages produced on the JES job log.
ABEND U(3142)	An error occurred processing the VIDIDCIN DD statement for IDCAMS. Review the additional diagnostic messages produced on the JES job log.

CICS transaction abend codes

Table 15. CICS transaction abend codes

Code	Header
ABEND VIDA	An error has occurred when attempting to access the system time of day clock.
ABEND VIDE	An attempt to load a CICS VT module has failed. Please contact IBM Technical Support.
ABEND VIDF	An invalid condition has occurred for a FREEMAIN command during CICS VT processing. Please contact IBM Technical Support.

Table 15. CICS transaction abend codes (continued)

Code	Header
ABEND VIDG	An invalid condition has occurred for a GETMAIN command during CICS VT processing. Please contact IBM Technical Support.
ABEND VIDH-Z	These abend codes indicate that an invalid response has been returned from a CICS command during CICS VT processing. Please contact IBM Technical Support.
ABEND 3139	DMF has detected a difference between the VSAM and DB2 record areas and the ERRORACT parameter in the DST definition for the file is ABEND. Review the diagnostics in the VIDCOMP DD statement.
ABEND 3140	DMF has detected a difference in the record lengths and the ERRORACT parameter in the DST definition for the file is ABEND. Review the diagnostics in the VIDCOMP DD statement.
ABEND 3141	DMF has detected a difference in the return code or reason code and the ERRORACT parameter in the DST definition for the file is ABEND. Review the diagnostics in the VIDCOMP DD statement.

DIM driver generation messages

Message ID	Message Text	Cause	Solution
VIDGD001	An attempt was made to generate DIM *ddddddd*, but no CICS VT mapping information exists for that DIM.	No VSAM to DB2 mapping information exists in the CICS VT system tables for the requested DIM.	Manually or automatically map the requested data. If manual mapping ensure that the DIM is marked as Ready for gen.
VIDGD002	SQL CODE *cccc* was encountered reading the CICS VT system table VID_DSN to generate DIM *ddddddd*.	The CICS VT generate drivers utility was unable to access the CICS VT system table VID_DSN.	Use the SQL code from the message and the accompanying formatted DB2 message in the job log to determine why the table is unavailable.
VIDGD003	No data set mapping information was found for the requested DIM *ddddddd*. Job terminated EID=*eeee*.	No VSAM to DB2 mapping information was found in the CICS VT system table VID_DSNMAP for the requested DIM.	Ensure that the requested DIM has been completely mapped using the CICS VT mapping facility.
VIDGD004	SQL CODE *cccc* received reading table VID_DSNMAP for DIM *ddddddd*, job terminated, EID=*eeee*.	The CICS VT generate drivers utility was unable to access the CICS VT system table VID.SEGM.	Use the SQL code from the message and the accompanying formatted DB2 message in the job log to determine why the table is unavailable.

Message ID	Message Text	Cause	Solution
VIDGD005	While attempting to generate DIM *ddddddd*, SQL CODE *cccc* was encountered reading system table VID_FIELD	The generate drivers utility was unable to access system table VID_FIELD	Use the SQL code from the message to determine why the table is unavailable.
VIDGD006	Error encountered while reading the field mapping information for the data set *ssssssss*, EID=*eeee*.	Either the CICS VT system table VID_FIELD is unavailable or a field is not mapped to CICS VT.	Use accompanying formatted DB2 message in the job log along with the diagnostic information following this to determine the error. If the SQL code is 100, verify that all of the fields for the data set have been mapped. If EID=E015, the first field in the CICS VT data set mapping is not defined as `KEY`.

Message ID	Message Text	Cause	Solution
VIDGD007	Error retrieving column information from SYSIBM.SYSCOLUMNS for *creator.table*, EID =*eeee*	The generate drivers utility was unable to retrieve information for the SYSIBM.SYSCOLUMNS table. This is because the SYSIBM.SYSCOLUMNS table is unavailable or a column name specified in the mapping does not exist.	For some errors, a formatted DB2 message will appear in the job log. Use that message along with the following diagnostic information to determine the cause of the problem. If the DB2 column name in the error message is the last column in the table, you have specified a DB2 column somewhere in the segment that does not exist in the DB2 table mapped to this segment. This can occur when you have loaded rows into VID_FIELD, rather than using the CICS VT ISPF mapping dialogs. **EID=001** A column specified in the mapping does not exist in the DB2 table for this DIM. Correct the column name and rerun the generate drivers utility. **EID= E017, E018, E021, E022, E024** Check the formatted DB2 message in the job log to determine the cause of the problem. **EID=E019, E020** Verify that the table in the message has been created.

Message ID	Message Text	Cause	Solution
VIDGD008	While attempting to generate DIM *dimname*, SQL CODE *cccc* was encountered checking the key field	Use the SQL CODE to determine cause of error. If SQL CODE is -811, then there is more than one key field defined.	Ensure that only one KEY field is defined and that CICS VT system tables are accessible.
VIDGD009	While attempting to generate DIM *dimname*, it was determined that the key field is greater than 255 bytes	The VSAM KEY defined to CICS VT is greater than 255 bytes.	Correct the mapping for the key.

DDM driver generation messages

Message ID	Message Text	Cause	Solution
VIDGP001	Invalid control statement, job terminated	An error has been found in the control statements that are input to stage 1 processing. The formats for the control statements are: **DIM=***dddddddd* Where *dddddddd* specifies the name of the DIM for which SQL drivers are to be built or **DDM=***pppppppp* Where *pppppppp* specifies the name of the DDM to be generated. There must be at least one of each type of statement in the stage 1 input.	This only occurs if you have edited the control statements generated by CICS VT. Correct the error and resubmit the batch generation job. If the control statements were generated by CICS VT, contact IBM Technical Support.
VIDGP002	Maximum number of DIMs or DDMs specified for generation. Run terminated.	The limit of 10000 DDMs or 1000 DIMs that can be generated at one time has been exceeded.	Specify more restrictive DDM selection criteria when selecting DDMs to be generated.
VIDGP003	SQL CODE *cccc* received deleting VID_SRC rows for data set *dddddddd*. Run terminated.	The generate drivers utility was unable to delete VID_SRC rows for the data set.	Use the SQL code from the message to determine why the delete failed.

Message ID	Message Text	Cause	Solution
VIDGP004	Unknown error routine called	The generate drivers utility was unable to process an error.	Send the dump produced by the generate utility to IBM Technical Support.
VIDGP005	SQL CODE *cccc* received while adding source to VID_SRC for DDM *pppppppp*. Job terminated.	Unable to add rows to the SQL drivers source table for the DDM named in the message.	Use the formatted DB2 message in the job log to determine the reason for the error. If the SQL code is -803, delete all the rows from the VID_SRC table where the column VIDSRPSB is the name of the DDM in the message, then generate the DDM again.
VIDGP006	Data set *ssssssss* not found in DIM *dddddddd*. Job terminated.	This is usually due to a mapping error.	Correct the CICS VT mapping for the DIM and retry this DDM SQL driver generation.
VIDGP007	Field *ffffffff* in data set *ssssssss* in DIM *dddddddd* not found in table VID_FIELD. Job terminated, EID=*eeee*.	A field defined in the DIM was not found in CICS VT system table VID_FIELD.	Use the formatted DB2 message in the job log to diagnose the problem.
VIDGP008	Data set *ssssssss* in DIM *dddddddd* not found in VID_DSNMAP table. Job terminated, EID=*eeee*.	No mapping information for the data set in the message could be found in CICS VT system table VID_DSNMAP.	Ensure that the DIM has been completely mapped.
VIDGP010	Error accessing SYSIBM.SYSCOLUMNS while looking up key columns for table *ttttttt*, EID=*eeee*	No key columns for the table named in the message could be found in SYSIBM.SYSCOLUMNS.	Ensure that the DB2 table specified for the data set primary table in the CICS VT mapping is correct and that the table has a primary index defined.
VIDGP012	An error has occurred while reading SYSIBM. SYSCOLUMNS when building the host variables for the SQL driver, EID=*eeee*.	The SYSIBM.SYSCOLUMNS table is unavailable, or the primary table DB2 definitions are incomplete.	Use the formatted DB2 message in the job log to determine why the table is unavailable. Ensure that the DB2 table definitions for the primary tables needed for the DIM are complete.

Message ID	Message Text	Cause	Solution
VIDGP017	An error was encountered trying to access SYSIBM.SYSCOLUMNS, EID=*eeee*.	There is a problem with the column information for the table.	Use supplementary information to determine the table in error and correct.
VIDGP018	Table *ttttttt*, column *cccccccc*, contains unsupported data type, EID=*eeee*	The DB2 data type for the column is not supported by CICS VT.	Change the data type for the column to one of the CICS VT supported data types: CHAR VARCHAR DECIMAL LONGVAR NUMERIC DATE SMALLINT BIGINT TIME INTEGER TIMESTAMP
VIDGP019	` SQL CODE *cccc* received accessing VID_SRC table for DIM *ddddddd*, data set *ssssssss*. Job terminated.	The CICS VT system table VID_SRC is not available.	Use the formatted DB2 message in the job log to determine why the table is unavailable.
VIDGP025	DDM *ddddddd* successfully processed.	The source code for the SQL drivers for the DDM has been successfully built in VID_SRC table. The stage 2 for the SQL drivers should now be run.	This is for information only.
VIDGP034	Unable to determine the primary key for the following primary DB2 table.	The GEN utility cannot process the DB2 primary key for the table associated with this data set.	Define the primary key and rerun the generation utility.
VIDGP038	SQL CODE *cccc* received reading table VID_FIELD for AIX *ddddddd*. Job terminated.	Mapping error for the alternate index path with a DIM name of *ddddddd*.	Correct the cause of the SQL error and rerun the generation utility.
VIDGP039	SQL CODE *cccc* received reading table SYSIBM.SYSCOLUMNS for AIX *ddddddd*. Job terminated.	Probable error in mapping of an alternate index path.	Correct the error and rerun the generation utility.

Message ID	Message Text	Cause	Solution
VIDGP041	SQL CODE *cccc* received verifying key length for DIM *dddddddd*. Job terminated.	An error has occurred with the mapped KEY field. If SQLCODE is -811 then there is more than one KEY field defined.	Correct the mapping for the KEY field.
VIDGP042	Data set *dddddddd* has a key greater than 255 bytes. Job terminated.	Unknown.	Correct the mapping for the KEY field.
VIDGP043	Error accessing DB2 bind member *name*. Member not found.	The member name specified in the defaults panel 1 for `BIND model in SVIDSAMP` does not exist in the SVIDSAMP data set.	Either create the member or specify an existing member.
VIDGP044	Error accessing DB2 bind member *name*. Return=`xx` Reason=`yy`.	There was a FIND macro error accessing member *name* in the defaults panel 1 for `BIND model in SVIDSAMP` option.	Review the FIND macro completion codes for `xx` and `yy` in *z/OS DFSMS Macro Instructions for Data Sets*.

CICS VT DB2 connection error messages

Message ID	Message Text	Cause	Solution
VIDCA001	DSNPARMS missing or invalid	There was an error reading the DSNPARMS parameter data set during a CICS VT batch utility function.	Correct the appropriate parameter data set. See “Creating CICS VT user data sets” on page 12 for further information.
VIDCA002	DB2 CONNECT error, subsystem=*ssss*, plan=*pppppppp*, return code=*rc*, reason code=*rsn*	The DB2 connect processing by the CICS VT ISPF component failed. *rc* and *rsn* are the codes returned by DB2. The most common reasons for this error is that the DB2 plan does not exist, or the user does not have sufficient authority to execute the plan.	Review the DB2 codes and take the appropriate action.
VIDCA003	DB2 OPEN error, subsystem=*ssss*, plan=*pppppppp*, return code=*rc*, reason code=*rsn*	The DB2 open processing by the CICS VT ISPF component failed. *rc* and *rsn* are the codes returned by DB2.	Review the DB2 codes and take the appropriate action.

Data migration generation utility messages

Message ID	Message Text	Cause	Solution
VIDMG001	Migration jobs successfully generated for DIM *dim-name*	This is an informational message.	No action required.
VIDMG002	Open failed for data set *data-set-name*	There was an error opening the data set *data-set-name.*	Review and correct the JCL.
VIDMG003	SQLCODE *ssss* returned for table *db2-table-name.*	There was an error accessing the DB2 table *db2-table-name*.	Review the SQL code in *DB2 for z/OS Codes* manual.
VIDMG004	SQLCODE *ssss* returned for table *db2-table-name*	There was an error accessing the DB2 table *db2-table-name*.	Review the SQL *ssss*code in *DB2 for z/OS Codes* manual.
VIDMG005	No mapping information found for DIM *dim-name*	No mapping information exists for the selected DIM *dim-name*.	Probable user error. Ensure that the DIM you select has been mapped and CICS VT drivers are generated.
VIDMG006	Error retrieving column count information for DIM *dim-name*	Error retrieving column count information for DIM *dim-name*.	An error occurred accessing the SYSIBM.SYSCOLUMNS table. Review the CAF messages written to the JES job log. If no other diagnostics are available, collect DB2 DSNTRACE output and contact IBM.
VIDMG007	DIM *dim-name* contains unsupported DB2 data type.	A column in the DB2 table for DIM *dim-name* is a data type that is not supported by CICS VT.	Generate the DIM to identify the unsupported column type and change the DDL to use only suppported column types.
VIDMG008	Error writing records for member *member-name*	An error was encountered creating the member name *member-name*.	Verify that the output data sets do not exist and rerun the data migration utility. If the problem persists, add a SYSDUMP DD statement and contact IBM.
VIDMG009	No input VSAM data set name detected or supplied	This is an error in the MIGPARM DD input statements.	Rerun the generation utility process from the CICS VT main menu. If the problem persists contact IBM.

Message ID	Message Text	Cause	Solution
VIDMG010	Data set name prefix for output data sets not supplied	This is an error in the MIGPARM DD input statements.	Rerun the generation utility process from the CICS VT main menu. If the problem persists contact IBM.
VIDMG011	Load failed for module VIDDDMEX	There was an error loading the DDM naming exit VIDDDMEX.	Add the library containing VIDDDMEX to the STEPLIB/JOBLIB.
VIDMG012	DIM *dim-name* is for alternate index path.	The specified DIM *dim-name* is for an alternate index.	The migration generation utility must have a base cluster DIM input.

CICS VT trace messages

Message ID	Message Text	Cause	Solution
VIDTR001	OPEN OF TRACE PARM DATA SET FAILED.	There was an error reading the parameter data set for CICS VT tracing. Processing continues without tracing.	Ensure that the VIDTRCEP DD statement is present and that it is an 80-byte fixed blocked sequential data set containing the parameters to be used for the trace.
VIDTR002	OPEN OF TRACE OUTPUT DATA SET FAILED.	The trace routines could not open the data set that is to contain the output from the trace. Processing continues without tracing active.	Ensure that the VIDTRCE DD statement is present and that it is a 133-byte fixed blocked sequential data set. If the VIDTRCD DD statement has been specified ensure that it is an 80-byte fixed blocked sequential data set.
VIDTR003	INVALID TRACE PARAMETER.	An invalid trace parameter was found in the VIDTRCEP data set.	Refer to for the correct format for the trace parameters.
VIDTR004	TRACE PARAMETERS MISSING.	No DIM names have been specified for tracing.	At least one DIM must be specified before tracing can be started. A maximum of ten DIM names may be specified at once for tracing.

CICS VT data conversion messages

Message ID	Message Text	Cause	Solution
VIDFC001	INVALID CONVERSION TYPE	An invalid field conversion routine has been supplied.	Correct the invalid conversion routine number in the parameter list.
VIDFC002	VIDCONV BAD PICTURE LENGTH	An invalid picture length has been supplied.	Correct the invalid picture length in the parameter list.
VIDFX001	INVALID DEFAULT DATA FOR NULL FIELD SETTING	An invalid null field default value has been supplied.	See CICS VT abend U3118.

CICS VT messages from VIDMAPIN

Message ID	Message Text	Cause	Solution
VIDIN001	OPEN FAILED FOR FILE ddddddd	The VIDMAPIN program was unable to open the data set name specified in the CATIN DD statement.	A VSAM data set name specified in the CATIN DD statement does not exist or could not be opened. Correct the invalid data set name and resubmit the job.
VIDIN002	ERROR RETURNED FROM SHOWCAT MACRO; RC=nn	The VIDMAPIN program has received an invalid return code from the SHOWCAT macro.	Review RC=nn for the SHOWCAT macro in the *DFSMS Macro Instructions for Data Sets* manual. If the problem persists, contact the IBM Technical Support.
VIDIN003	ERROR READING SHOWCAT MACRO OUTPUT	The VIDMAPIN program is unable to read the transient data set produced by the SHOWCAT MACRO.	The SHOWCAT macro output is written to the SYSPRINT DD. The default device type is VIO. Change this to SYSDA and rerun. If the problem persists, contact IBM Technical Support.
VIDIN004	ERROR RETURNED FROM THE LISTCAT MACRO; RC=nn.	The VIDMAPIN program has received an invalid return code from the LISTCAT macro.	Review RC=nn for the LISTCAT macro in the *DFSMS Macro Instructions for Data Sets* manual. If the problem persists, contact the IBM Technical Support.

Message ID	Message Text	Cause	Solution
VIDIN005	SQLCODE ssss RETURNED WHEN PERFORMING THE SQL OPERATION aaaaaa.	The VIDMAPIN has encountered an SQL code ssss during an SQL call aaaaaa. Possible values of aaaaaa are INSERT or DELETE.	Correct the DB2 problem and resubmit the batch job.

CICS VT batch subsystem messages

Message ID	Message Text	Cause	Solution
VIDSS200	SUBSYSTEM NAME=*ssss* OPENING DDNAME=*yyy*	The VT batch subsystem *ssss* has opened the DDNAME *yyy*.	This message is issued for information.
VIDSS201	SUBSYSTEM NAME=*ssss* CLOSING DDNAME=*yyy*	The VT batch subsystem *ssss* has closed the DDNAME *yyy*.	This message is issued for information.
VIDSS202	SUBSYSTEM NAME=*ssss* HIT MAXIMUM EXPECTED ACBs FOR DDNAME=*yyy*	The VT batch subsystem *ssss* has attempted to open an ACB for DDNAME *yyy*. VT supports a maximum of 10 ACBs for one VSAM DD.	Modify the application program to reduce the number of ACBs for a single DD.
VIDSS203	SUBSYSTEM NAME=*ssss* HIT MAXIMUM EXPECTED RPLs FOR DDNAME=*yyy*. VT supports a maximum of 100 RPLs for one VSAM DD	The VT batch subsystem *ssss* has attempted to open an RPL for DDNAME *yyy*. VT supports a maximum of 100 RPLs for one VSAM DD.	Modify the application program to reduce the number of RPLs for a single DD.
VIDSS204	MODULE NAME= *lmod* COULD NOT BE LOADED	The VT batch subsystem is unable to load the load module *lmod*.	Ensure that the library containing the required load module is available in STEPLIB/JOBLIB or in a library in the MVS link list. If the module name is *VIDFMSP*, a DD statement specifying the VT subsystem also has a DSN= parameter and you are not executing version 2.1 code.

Message ID	Message Text	Cause	Solution
VIDSS205	REQUIRED SUBSYSTEM PARAMETER NOT GIVEN ON DDNAME=*yyy* FOR SUBSYSTEM NAME=*ssss*	Too few parameters are specified for the VT subsystem *ssss* and DD name *yyy*. As a minimum, the subsystem requires the DB2 *ssid* and the DIM name.	Correct the invalid parameter list.
VIDSS206	POSITIONAL PARAMETER NUMBER *n* TOO LONG ON DDNAME=*yyy* FOR SUBSYSTEM NAME=*ssss*	The length of the *n*th parameter for the subsystem *ssss* and DD name *yyy* exceeds the maximum length of 8 bytes.	Correct the invalid parameter number *n* and resubmit.
VIDSS207	DB2 SUBSYSTEM NAME NOT GIVEN ON DDNAME *yyy* FOR SUBSYSTEM NAME *ssss*	The second input parameter for DDNAME *yyy* and subsystem *ssss* does not conform to the DB2 rules for subsystem name.	Correct the 2nd positional parameter to specify a valid DB2 subsystem and resubmit.
VIDSS208	NO PARAMETERS SPECIFIED IN JCL ON DDNAME=*yyy* FOR SUBSYSTEM NAME=*ssss*	The parameter list for subsystem *ssss* and DD name *yyy* does not specify the minimum DB2 *ssid* and DIM name.	Correct the parameter list and resubmit.
VIDSS209	SUBSYSTEM PARAMETERS DB2 SUBSYSTEM *ssid* DRIVER *dim-name* EXIT NAME *exit-name*	This is an information only message that is generated when additional VT diagnostics are being captured.	This message is issued for information.
VIDSS210	MODULE NAME= *lmod* LOCATED AT EPA *addr*	This is an information only message that is generated when additional VT diagnostics are being captured.	This message is issued for information.
VIDSS211	CONTROL BLOCK *name* LOCATED AT *addr*	This is an information only message that is generated when additional VT diagnostics are being captured.	This message is issued for information.
VIDSS212	CONCATENATED FILES ARE NOT SUPPORTED, TERMINATING	A DD statement specifying the SUBSYS parameter is part of a concatenation. This is not supported by VT.	This message is issued for information.

Message ID	Message Text	Cause	Solution
VIDSS213	I/O STATISTICS FOR DDNAME *yyy*	This is an information only message that is generated when additional VT diagnostics are being captured.	This message is issued for information.
VIDSS214	I/O OPERATION=*call-type*, COUNT=*n*	This is an information only message that is generated when additional VT diagnostics are being captured.	This message is issued for information.
VIDSS215	MODULE EYECATCHER: *lmod*	This is an information only message that is generated when additional VT diagnostics are being captured.	This message is issued for information.
VIDSS216	OPTION TABLE OBTAINED VIA *location*	This is an information only message that indicates whether the subsystem options table was loaded from STEPLIB/JOBLIB or from the LPA.	This message is issued for information.
VIDSS217	OPTION TABLE FIELD *field* HAS VALUE *x*	This is an information only message generated when additional VT batch subsystem diagnostics are being captured. The message is also issued when program VIDSSUTL is executed with the REPORT option. In the messages, *x* shows the value of the option table fields.	This message is issued for information.
VIDSS218	USING MESSAGE TABLE NAME=*table*	This is an information only message that is generated when additional VT diagnostics are being captured.	This message is issued for information.

Message ID	Message Text	Cause	Solution
VIDSS219	DUMP OF *aaa* RECORD	This is an information only message that is generated when additional VT diagnostics are being captured. *aaa* is either DB2 or VSAM.	This message is issued for information.
VIDSS220	*Addr*	This is an information message issued in conjunction with VIDSS219 and VIDSS227.	This message is issued for information.
VIDSS221	NAME/TOKEN CREATE RC=*rc* FOR *control block*	This is an information only message that is generated when additional VT diagnostics are being captured.	This message is issued for information.
VIDSS222	SUBSYSTEM PARAMETERS DB2 SUBSYSTEM . . . *ssid* DRIVER . . . . . *driver* VSAM DDNAME . . . *yyy* EXIT. . . . . *vt-exit*	This is an information only message that is generated by DMF.	This message is issued for information.
VIDSS223	VSAM DDNAME= *yyy* NOT ALLOCATED SUBSYS=*ssss*	DMF is unable to open the DD name *yyy* for the VSAM dataset.	Ensure that the same name is specified as the DMF VSAM data set subsystem parameter and in the JCL for a batch job and in the CICS FCT.
VIDSS224	VSAM DDNAME=*yyy* DOES NOT POINT TO A VSAM FILE, SUBSYS=*ssss*	This message is issued by DMF and means that the DD name specified as the fourth parameter in the SUBSYS statement does not reference a valid VSAM data set.	Correct the SUBSYS parameter or specify a valid VSAM data set for *yyy*.
VIDSS225	VSAM DDNAME=*yyy* IS A CONCATENATION, SUBSYS=*ssss*	This message is issued by DMF and means that the DD name specified in the fourth SUBSYS parameter is a concatenation. This is not supported by VT.	Correct the DD name *yyy*.

Message ID	Message Text	Cause	Solution
VIDSS226	DUAL MODE ERROR FOR SUBSYSTEM *ssss* VSAM DDNAME *yyy* RLEN NOT SAME: DB2=*db2-len* VSAM=*vsam-len* FROM GET	DMF has detected a difference in the DB2 and VSAM record lengths.The respective record lengths are *db2-len* and *vsam-len*.	There are two typical reasons for this error: 1. The VSAM file contains varying length records but is mapped as fixed length (KSDS or RRDS). 2. The record are is being built incorrectly by a user exit.
VIDSS227	DUAL MODE ERROR FOR SUBSYSTEM *ssss* VSAM DDNAME *yyy* RECORD DATA WAS NOT THE SAME AT OFFSET X'*nn*'	DMF has detected a difference in the data retrieved from DB2 and VSAM. The first data difference is at offset X' *nn*'.	This may be an issue with the synchronization of the data in DB2 and VSAM. Review the console messages.
VIDSS228	DUAL MODE ERROR FOR SUBSYSTEM *ssss* VSAM DDNAME *yyy* FEEDBACK DIFFERENTIATION *call-type* DB2 RPLRTNCD=*nn* RPLERRCD=*nn* VSAM RPLRTNCD=*nn* RPLERRCD=*nn* ACB MACRF=(*n1,n2*) RPL OPTCD=(*n3,n4*) KEYL=*nnnn*	DMF has detected a difference in the return codes from DB2 and VSAM	This may be an issue with the synchronization of the data in DB2 and VSAM. Review the console messages.
VIDSS229	VSAM DDNAME *yyy* NOT FOUND IN TIOT, SUBSYS=*ssss*	This message is issued by DMF and means that the DD name specified in the fourth SUBSYS parameter was not found in the job JCL.	Add the appropriate DD statement for *yyy*.
VIDSS230	VSAM DDNAME *yyy* JFCB ERROR, SUBSYS=*ssss*	The JFCB for DD name *yyy* could not be located.	Notify IBM support.
VIDSS231	VSAM DDNAME *yyy* DSCB NOT FOUND, SUBSYS=*ssss*	The obtain for the format 1 DSCB failed. This can occur when the VSAM file is catalogued but does not reside on the volume in the catalog entry.	Verify that the data set is on the volume where it is catalogued.
VIDSS232	BUFFER OBTAINED=*n1* FOR RPL=*n2* SUBSYS *ssss*	This is an information only message that is generated by DMF	This message is issued for information.
VIDSS233	DB2 DDNAME= *yyy*, RC =*n1*, RPLRTNCD=*n2*, RPLERRCD=*n3*	This is an information only message that is generated by DMF	This message is issued for information.

Message ID	Message Text	Cause	Solution
VIDSS234	VSAM DDNAME= *yyy*, RC =*n1*, RPLRTNCD=*n2*, RPLERRCD=*n3*	This is an information only message that is generated by DMF	This message is issued for information.
VIDSS235	VSAM DDNAME=*yyy*, OPEN RC= *n1*	DMF has encountered return code *n1* on an OPEN for the VSAM DD name *yyy*.	Review the accompanying IEC141I message.
VIDSS236	IGNORING FEEDBACK DIFFERENCE FOR SPECIAL CASE: VSAM RPLRTNCD=*n1* RPLERRCD=*n2*	This message is issued in diagnostic mode, and indicates that DMF has detected a difference with RPLERRCD but is tolerating this error due to the existence of the VIDSSRCx DD statement.	This message is issued for information.
VIDSS237	IGNORING DDNAME= *yyy*, 8th CHARACTER is NOT 0, 4, 8 OR A	The VIDSSRCx DD statement has been incorrectly specified.	Correct the invalid statement.
VIDSS238	NUMBER OF VSAM ERRORS IGNORED=*n1* FOR SPECIAL TYPE=*n2*	Special DD *n2* (VIDSSIO OR VIDSSRC) was coded and this message shows the total number of VSAM errors.	This message is issued for information.
VIDSS239	ACB FOR APPLICATION AT OPEN FOUND AT *addr*.	This diagnostic message shows the address in storage for the ACB.	This message is issued for information.
VIDSS240	ACB FOR APPLICATION AT OPEN WAS NOT FOUND *reason*.	The primary reason this message is issued is that when the program issuing the open for the file used a DCB instead of an ACB. A secondary reason for this message is that the logic to find the ACB in storage was unsuccessful. In this case, contact IBM support.	This message is issued for information.

Message ID	Message Text	Cause	Solution
VIDSS250	SUBSYSTEM NAME=*ssid* VSAM ONLY FOR DDNAME=*ddname*	This is for information only and requires no action. The DD statement *ddname* has been updated to enable VT access in subsystem *ssid* and the DIM specified in the SUBSYS JCL parameter is defined in the VT exceptions module VIDFMSPX.	This message is issued for information.

Appendix A. CICS VT transactions

The following table lists all of the transactions that are supplied with CICS VSAM Transparency. All of the transactions can be found in the VID CSD group.

Table 16. List of CICS VT transactions

Transaction	Security category	Description
VTMA	2	Starts and stops the dual mode facility (DMF).
VTMC	2	Checks that the CICS installation was successful.
VTMD	2	Displays the list of files that have been migrated and their current status.
VTMI	2	Starts and stops the CICS VT interface and reloads the DST.
VTMM	2	Changes the migration status of an individual file.
VTMT	2	Activates tracing in the CICS region.
VTMU	2	Provides CICS VT usage information.
VTMV	2	Runs the installation verification procedure.

Appendix B. CICS VT utilities and samples

The following tables list all of the utilities and sample JCL that are supplied with CICS VSAM Transparency. You run the CICS VT utilities using the sample JCL to set up CICS VT correctly and migrate data sets to DB2.

CICS VT provides sample JCL to help you correctly set up and configure the product. This JCL is in the *my* .SVIDCNFG. *cust* library.

Table 17. Sample JCL to set up CICS VT

JCL	Description
VIDCDEF	Defines the CICS VT definitions to the CSD using the DFHCSDUP utility.
VIDCDEF2	The new version 2.1 objects to be defined in the CICS CSD. It is used if you are upgrading an existing version 1.2 installation.
VIDDEF	Creates the CICS VT parameter data sets.
VIDGDMEX	Assemble/link JCL and source for VIDDDMEX exit.
VIDSETUP	Runs the CICS VT installation customization program.

CICS VT provides a number of utilities to prepare data sets for migration to DB2. The sample JCL to run these utilities are in the *my.* SVIDSAMP. *cust* library. Some of the sample JCL members have the same name as the utilities that they execute.

Table 18. Utilities and sample JCL for data migration

CICS VT Utility	JCL name	Utility description
VIDMAPIN	VIDAUTOJ	Gathers information about one or more VSAM data sets for the automated mapping facility.
VIDFCTST	VIDCONVT	Tests the default data conversion routines.
VIDGDTAB	VIDGDTAB	Assembles the CICS VT data set table that contains the list of migrated data sets.
VIDLOAD	VIDLOAD	Creates the DB2 load data set.
VIDLOADV	VIDLOADV	Falls back to VSAM.
VIDPATHM	VIDPATHJ	Manually maps alternate indexes.
VIDUNLOD	VIDUNLOD	Unloads a VSAM file to a sequential data set, prior to converting the data to DB2 format.
VIDMIGGN	None	Generates the data migration jobs and control information
VIDGFMX	VIDGFMX	Source code and assemble/link JCL for VIDFMSPX

To help you migrate your data sets to DB2, CICS VT provides sample JCL that you can edit to run DB2 utilities. The sample JCL to run these utilities are in the *my* .SVIDSAMP. *cust* library.

Table 19. Sample JCL for DB2 utilities

DB2 utility	Sample JCL	Description
REPAIR	VIDB2RPR	Resets the image copy pending a flag in DB2 after using the LOAD utility with `LOG NO` specified.
LOAD	VIDB2LD	Loads the converted data to DB2.
RUNSTATS	VIDRUNS	Updates the DB2 catalog systems after the data is loaded to DB2.
DSNTIAUL	VIDTIAUL	Generates the control cards for the DB2 LOAD utility.
DSNTEP2	VIDTEP2	Runs the dynamic SQL in batch mode.

Glossary

access method control block (ACB)
A control block that links an application program (for example, a CICS system) to VSAM or VTAM.

alternate index
For VSAM key-sequences data sets and entry-sequenced data sets, an index of alternate keys that provides a path for secondary access to the data set.

bind The process by which the output from the SQL precompiler is converted to a usable control structure, often called an access plan, application plan, or package. During this process, access paths to the data are selected and some authorization checking is performed.

buffer pool
Main storage that is reserved to satisfy the buffering requirements for one or more DB2 table spaces or indexes.

call attachment facility (CAF)
A DB2 attachment facility for application programs that run in TSO or MVS batch.

copybook
The definition of a record structure that describes the record at a field level to an application program.

data definition language (DDL)
A set of SQL statements used to create DB2 objects.

data migration component
The CICS VT component that is used to perform the initial data migration from VSAM to DB2.

database request module (DBRM)
A data set member that is created by the DB2 precompiler and that contains information about SQL statements. DBRMs are used in the bind process.

data set definition module (DDM)
A module that is used by CICS VT at run time to access the migrated data in DB2. One DDM is created for each base cluster and alternate index.

data set information module (DIM)
A module that defines the relationship between the VSAM record structure and the DB2 table structure. One DIM is created for each base cluster and alternate index path.

data set table (DST)
A CICS table that contains definition for all the VSAM data sets that are migrated using CICS VT.

entry-sequenced data set (ESDS)
A VSAM data set whose records are physically in the same order in which they were put in the data set. It is processed by addressed direct access or addressed sequential access and has no index. New records are added at the end of the data set.

field build exit (FBE)
A user written program that CICS VT uses for field level data reengineering.

global user exit (GLUE)
A point in a CICS module at which CICS can pass control to a user-written program (known as an exit program), and then resume control when the program has finished. CICS VT uses a GLUE to intercept VSAM calls issued by programs running in CICS.

installation verification procedure (IVP)
The series of post-configuration tasks that are used to verify that the set up and configuration of CICS VT has been successful.

insert, replace delete exit (IRD)
A user written program that CICS VT uses to perform record level reengineering.

Interactive System Productivity Facility (ISPF)
An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, it provides a means of generating standard screen panels and interactive dialogs between the application programmer and terminal user.

job control language
A control language that is used to describe a job and its requirements to an operating system.

Job Entry Subsystem (JES)
An IBM licensed program that receives jobs into the system and processes all output data that is produced by jobs.

key-sequenced data set
In a z/OS environment, a VSAM file or data set whose records are loaded in key sequence and controlled by an index.

library lookaside (LLA)
A facility in MVS/ESA that reduces library I/O activity by keeping selected directory entries in storage, instead of making repetitive searches of DASD.

mapping
A CICS VT process that establishes the relationship between a VSAM record and a DB2 row, in order to generate DIM and DDM run time modules.

mapping component
The CICS VT ISPF component that is used to perform mapping.

migration unit
A number of VSAM data sets that are migrated at the same time.

page set
A table space or index space. Each page set consists of a collection of VSAM data sets.

primary index
An index that enforces the uniqueness of a primary key.

primary key
In a relational database, a unique, non-null key that is part of the definition of a table. A table cannot be defnied as a parent unless it has a unique key or primary key.

program list table (PLT)
The CICS control table that contains a list of programs. The programs in a PLT can be executed as a group during CICS start up or shutdown, and can be enabled and disabled as a group by a single CEMT transaction.

relative record data set (RRDS)
A VSAM data set organization, in which records are of fixed length and are accessed by their relative record numbers. The relative record number (RRN) of a record is its displacement (in records) from the beginning of the data set.

relative record number (RRN)
In an RRDS, the number of the "slot" used to hold a record, that is its displacement (in records) from the beginning of the data set.

run-time component
The component of CICS VT that intercepts VSAM calls issued by application programs and processes them in DB2.

SQL Processor Using File Input (SPUFI)
A facility of the TSO attachment subcomponent that enables the DB2I user to execute SQL statements without embedding them in an application program.

storage group
A named set of disks on which DB2 data can be stored.

subsystem interface (SSI)
The MVS interface by which routines request services of, or pass information to, subsystems. The SSI is used by CICS VT to intercept VSAM calls issued by batch programs.

task-related user exit (TRUE)
A user exit program that is associated with specified events in a particular task, rather than with every occurrence of a particular event in CICS processing (as is the case with global user exits).

Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully.

You can perform most tasks required to set up, run, and maintain CICS VSAM Transparency in one of these ways:

- Using a 3270 emulator logged on to CICS
- Using a 3270 emulator logged on to TSO
- Using a 3270 emulator as an MVS system console

IBM Personal Communications provides 3270 emulation with accessibility features for people with disabilities. You can use this product to provide the accessibility features you need for CICS VSAM Transparency.

Index

Numerics

A

B

C

D

E

F

G

I

J

K

L

M

N

O

P

R

S

T

U

V

Z

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.

Readers' Comments — We'd Like to Hear from You

CICS VSAM Transparency for z/OS
Version 2 Release 1
User's Guide

Publication No. SC34-7249-00

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:
- Send your comments to the address on the reverse side of this form.
- Send a fax to the following number: +44 1962 816151
- Send your comments via email to: idrcf@uk.ibm.com

If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	Email address

Readers' Comments — We'd Like to Hear from You
SC34-7249-00

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM United Kingdom Limited
User Technologies Department (MP189)
Hursley Park
Winchester
Hampshire
United Kingdom
SO21 2JN

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

SC34-7249-00