



SERENA[®]
StarTool[®] DA 5.7.2

Batch User's Guide

Serena Proprietary and Confidential Information

Copyright

Copyright © 2001-2011 Serena Software, Inc. All rights reserved.

This document, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by such license, no part of this publication may be reproduced, photocopied, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Serena. Any reproduction of such software product user documentation, regardless of whether the documentation is reproduced in whole or in part, must be accompanied by this copyright statement in its entirety, without modification.

This document contains proprietary and confidential information, and no reproduction or dissemination of any information contained herein is allowed without the express permission of Serena Software.

The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Serena. Serena assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

Trademarks

Serena, PVCS, Mariner, Dimensions, ChangeMan, Comparex, StarTool, TeamTrack, and Collage are registered trademarks of SERENA Software, Inc. Mashup Composer, Meritage and Prototype Composer are trademarks of SERENA Software, Inc.

All other products or company names are used for identification purposes only and may be trademarks of their respective owners. Nothing contained in this document should be construed as granting any license or right to use any trademark without the prior written permission of the party that owns the trademark.

U.S. Government Rights

Any Software product acquired by Licensee under this Agreement for or on behalf of the U.S. Government, its agencies and instrumentalities is "commercial software" as defined by the FAR. Use, duplication, and disclosure by the U.S. Government is subject to the restrictions set forth in the license under which the Software was acquired. The manufacturer is Serena Software, Inc., 1900 Seaport Boulevard, 2nd Floor, Redwood City, California 94063-5587.

Publication date: June 2011

Table of Contents

	Welcome to Serena StarTool DA Batch	7
	Before You Begin	7
	Conventions	7
	Documentation	8
	Accessing the Documentation	8
	Using the PDF Documentation	9
<i>Chapter 1</i>	Introduction	11
	Conceptual Overview	12
	Component Overview	13
<i>Chapter 2</i>	Debugging and Viewing Server	17
	Starting a DVS Session	18
	Selecting a Dump Data Set	19
	Selecting Options from the DVS Selection Menu	19
	Processing a Dump Data Set with IPCS	20
	Deleting a DUMP Data Set	21
	Deleting a Source Listing from the Language Database	22
	Using the DVS Panels	22
	Option 1 – System and Job Information	24
	Option 2 – Diagnostic Section	24
	Option 3 – Failing Instruction Section	26
	Option 4 – Register and PSW Information	27
	Option 5 – Program Call Trace Summary	32
	Option 6 – Program Storage	37
	Commands for the Storage Display Panel	38
	Option 7 – Program Listing (COBOL only)	42
	Option 8 – Data Management Control Blocks and Buffers	45
	Option C - ChangeMan ZMF Information	47
	Option D - Program Call/Trace Details	49
	Option L - Load a source listing to DA	49
	Multilevel Storage Viewing	52
	Program Call/Trace Summary for Inactive Modules	54
<i>Chapter 3</i>	Mini Dump Debug Report	55
	Mini Dump Sections	56
	Mini Dump Report Example	57
	Job and System Section	57
	Diagnostic Section	58
	Failing Instruction Section	59
	Program Status Word and Registers Section	61
	Call Trace Section	63

	COBOL Data Section	64
	Data Management Control Blocks Section.	66
	ChangeMan ZMF Information Section.	68
<i>Chapter 4</i>	COBOL Source-Level Support	69
	Processing Modes.	70
	Production Mode	70
	Development Mode	71
	Differences Between Production and Development	71
	Language Environment.	72
	Invoking StarTool DA-Batch within IBM's Language Environment . . .	72
	Debugging and Viewing Server Call/Trace Summary for COBOL Programs	73
<i>Chapter 5</i>	DB2 Option	75
	DB2 Diagnostics	76
	SQL Call Summary Panel	76
	Plan Information Panel	77
	Package Information Panel	80
	DBRM Information Panel.	84
	SQL Statement and Host Variables Panel	86
<i>Chapter 6</i>	IMS Option	89
	What is Supported	90
	Information Displayed	91
	Viewing a Dump with DVS	92
	Point-and-Shoot	94
	Relocated PCBs	96
	IMS Region Abends	97
<i>Chapter 7</i>	ChangeMan ZMF Option	99
	Selecting the Dump to Analyze	100
	General Package Information Panel	101
	Load Component Information Panel	102
	History Information Panel	103
	Display ChangeMan Source Listings	104
	Logic for Retrieving Source Code from ChangeMan	105
<i>Chapter 8</i>	Dump Examples	107
	SOCB with Language Server (COBOL II).	108
	S130 Abend	112
	S813 (Diagnostic Section Only)	115
	SOC4 - 11 (Storage Not Available).	116
	SOC4 - 4 (Low Storage Access)	118
	SOC4 - 4 (Storage Key Mismatch)	119
	SOC4 - 4 (Storage Protected)	120
	DB2 Example	121

Chapter 9

Troubleshooting	123
Missing Dump Display Options	124
No Debugging and Viewing Server Dumps Available	124
No Mini Dump Debugging Reports Available	124
COBOL Source Display	124
No COBOL source in Mini Dump or Debug/Viewing Server	124
Wrong Version of COBOL Source in the Debug/Viewing Server	125
Failing COBOL Statement Not Displayed	125
ChangeMan ZMF Integration	125
StarTool DA Fails to Connect with ChangeMan ZMF	125
Index	127

Welcome to Serena StarTool DA Batch

Product Description Thank you for choosing Serena® StarTool® DA (Dump Analyzer), the intelligent dump analysis and debugging tool for IBM z/OS mainframes. StarTool DA helps mainframe software engineers diagnose the causes of **abnormal end**-of-job (ABEND) conditions. The product provides automated dump diagnostics, interactive call tracing support, dump archiving and maintenance, and a full-featured debugging interface.

Batch, CICS, DB2, and IMS

StarTool DA supports analysis and debugging of both batch and CICS ABENDs. Optional debugging support for IBM DB2 and IMS database ABENDs is also available.

ChangeMan ZMF

Users of Serena® ChangeMan® ZMF may optionally integrate their software change management system with the debugging support of StarTool DA using StarTool DA's ZMF Integration Option.

Document Objectives The *StarTool DA Batch User's Guide* manual provides information about using the interactive debugging features of StarTool DA Batch to assist with the diagnosis and resolution of mainframe batch program ABENDs.

Audience This information is intended for mainframe programmers who use StarTool DA as an ABEND debugging tool. Familiarity with IBM mainframe computer systems and IBM mainframe COBOL and assembly language programming is assumed.

Before You Begin

See the Readme for the latest updates and corrections for this manual. You can download the latest version of the Readme from the [Serena support website](#).

Conventions

- Terminological** The following terminological and style conventions are used throughout this document:
- z/OS™ represents the z/OS and OS/390® IBM® operating systems.
 - StarTool DA Batch also is referred to as CICS Dump Analyzer or DA CICS.
 - Examples may show either StarTool DA or ChangeMan ZMF-DA in screen and print titles. The product name in the title indicates whether the StarTool DA installation is configured to integrate with ChangeMan ZMF, and both types of installations were used in the preparation of examples for this manual.
- Typographical** The following typographical conventions are used in the online manuals and online help. These typographical conventions are used to assist you when using the documentation;

they are not meant to contradict or change any standard use of typographical conventions in the various product components or the host operating system.

Convention	Explanation
<i>italics</i>	Introduces new terms that you may not be familiar with and occasionally indicates emphasis.
bold	Emphasizes important information and field names.
UPPERCASE	Indicates keys or key combinations that you can use. For example, press the ENTER key.
monospace	Indicates syntax examples, values that you specify, or results that you receive.
monospaced <i>italics</i>	Indicates names that are placeholders for values you specify; for example, filename.
vertical rule	Separates menus and their associated commands. For example, select File Copy means to select Copy from the File menu. Also, indicates mutually exclusive choices in a command syntax line.

Documentation

StarTool DA publications include the Readme file and the following:

Title	Description
<i>Serena[®] SER10TY User's Guide</i>	Installation information for SER10TY licensing software and instructions on how to apply license key SERTificates.
<i>Serena[®] StarTool[®] DA Batch User's Guide</i>	Concepts, features, and functions of StarTool DA Batch, which provides dump analysis and ABEND recovery assistance in a batch environment.
<i>Serena[®] StarTool[®] DA CICS User's Guide</i>	Concepts, features, and functions of StarTool DA CICS, which provides dump analysis and ABEND recovery assistance in a CICS environment.
<i>Serena[®] StarTool[®] DA Messages</i>	A consolidated message reference for all product options, with error recovery recommendations.
<i>Serena[®] StarTool[®] DA Installation and Setup</i>	A guide on how to install and configure StarTool DA Batch and StarTool DA CICS.
<i>Serena[®] StarTool[®] DA Batch Training Guide</i>	A quick-start guide to StarTool DA Batch basic features and functions.
<i>Serena[®] StarTool[®] DA CICS Training Guide</i>	A quick-start guide to StarTool DA CICS basic features and functions.

Accessing the Documentation

The *StarTool DA Installation and Setup* manual and the *Serena SER10TY User's Guide* are shipped with the product as printed manuals and are included on the product media. In addition, the full documentation suite is provided in electronic form on the Serena Support website.

You may copy the electronic documentation to an intranet server for internal use, subject to the terms of the Master License and Services Agreement (MLSA).

Accessing Documentation Online

All product documentation is available for download to licensed customers on the Serena Customer Support website at <http://support.serena.com>. A user ID and password are required to log on to the website.

- 1 After logging onto the site, click on the **My Downloads** tab.
- 2 From the list of products in the **Please Select Product** drop-down menu, select StarTool DA.
- 3 In the download list, find the desired version of StarTool DA, then click the **Download Documentation** link for that version.

A page displays a list of available documentation for the selected release.

- 4 Click the **Download** link for the document you want to download.

Using the PDF Documentation

To view PDF files, use Adobe® Reader®, which is freely available from www.adobe.com.



TIP Be sure to download the *full version* of Reader. The more basic version does not include the search feature.

This section highlights some of the main Reader features. For more detailed information, see the Adobe Reader online help system.

The PDF manuals include the following features:

- **Bookmarks.** All of the manuals contain predefined bookmarks that make it easy for you to quickly jump to a specific topic. By default, the bookmarks appear to the left of each online manual.
- **Links.** Cross-reference links within a manual enable you to jump to other sections within the manual and to other manuals with a single mouse click. These links appear in blue.
- **Printing.** While viewing a manual, you can print the current page, a range of pages, or the entire manual.
- **Comments.** All PDF documentation files that Serena delivers with ChangeMan ZMF have enabled commenting with Adobe Reader. Adobe Reader version 7 and higher has commenting features that enable you to post comments to and modify the contents of PDF documents. You access these features through the Comments item on the menu bar of the Adobe Reader.
- **Advanced search.** Starting with version 6, Adobe Reader includes an advanced search feature that enables you to search across multiple PDF files in a specified directory. (This is in addition to using any search index created by Adobe Catalog—see step 3 below.)

To search within multiple PDF documents at once, perform the following steps (requires Adobe Reader version 6 or higher):

- 1** In Adobe Reader, select **Edit > Search** (or press **CTRL+F**).
- 2** In the text box, enter the word or phrase for which you want to search.
- 3** Select the **All PDF Documents in** option, and browse to select the folder in which you want to search.
- 4** Optionally, select one or more of the additional search options, such as **Whole words only** and **Case-Sensitive**.
- 5** Click the **Search** button.



NOTE Optionally, you can click the **Use Advanced Search Options** link near the lower right corner of the application window to enable additional, more powerful search options. (If this link says **Use Basic Search Options** instead, the advanced options are already enabled.) For details, see Adobe Reader's online help.

Chapter 1

Introduction

StarTool DA is a dump management system that provides detection, capture, notification, diagnostics, and analysis for application abends. It is specifically engineered to provide universally accepted IBM dumps, such as CICS formatted transaction dumps and IBM SVC (SDUMP) dumps, so that you can eliminate the process of production problem replication.

StarTool DA has a flexible interface to integrate with the z/OS™ and OS/390® IBM® operating systems. All dumps normally requested by z/OS or OS/390 are supported by StarTool DA.

StarTool DA is easy to install, use, and maintain. With a minimum learning curve, it can help you improve the reliability of your systems and increase the productivity of systems and application programmers.

Conceptual Overview	12
Component Overview	13

Conceptual Overview

StarTool DA captures information on the abending program and the system and produces applicable diagnostics in various formats.

If an abend occurs and a SYSUDUMP, SYSMDUMP, or SYSABEND DD is present, the operating system creates a dump. Alternatively, if an abnormal situation occurs within the z/OS Language Environment, control is passed to an LE exit that StarTool DA replaces. Under either condition, StarTool DA gets control and dynamically loads its modules to process the abend.

StarTool DA operates with default parameters, or you can customize them to suit your needs. You can change selection values such as the specific abend code, programming language, file access method, and MVS Control Table options. StarTool DA operates according to the MVS Control Table used at start-up.

Modes of Operation for COBOL

StarTool DA operates in Development Mode and Production Mode. You can use them interchangeably in the development and production environments. For both modes, the program must be compiled with the SOURCE, LIST, MAP, XREF, and NOOFFSET options.

Development mode

Development mode is a mode in which StarTool DA considers the abend as having come from a program that is still in development. This is defined by having the following DD statement in the execution JCL:

```
//ESPYCOBL DD DSN=xxxxxxxx
```

where xxxxxxxx is the name of the PDS (without the member name) or sequential data set that contains the compile listing to be used by StarTool DA. The compile listing is the SYSPRINT output from the compile step. After-the-abend program compiles are not supported.

Production mode

Production mode is a mode in which StarTool DA considers the abend as having come from a program that is in production. This is defined as all jobs that do not have an ESPYCOBL DD statement and those that pass the RAChecks within the StarTool DA product. Your StarTool DA Batch system administrator must add two StarTool DA Batch language processor steps to your compile JCL.

Types of Dumps

Three types of dumps are available with StarTool DA. You can produce any one or a combination of the following dumps by using the StarTool DA MVS Control Table (EMCT) or pre-defined DD names. The three types of dumps are:

- Mini Dump – The Mini Dump Debug Report includes only the essential information you need to diagnose and solve the abend. The information is specific to the type of abend, the language of the failing program, and the types of files or databases in use. The report also shows the failing COBOL statement and the contents of the operands.
- IBM Formatted Dump – This is the traditional IBM SYSUDUMP, SYSABEND, or SYSMDUMP. These dumps are only produced when their DD names are detected in the

abending step. Many vendors require one of these dumps before attempting any error analysis.



NOTE StarTool DA does not produce diagnostic information unless it detects the presence of one of these DD names: SYSUDUMP, SYSABEND, SYSMDUMP.

- Debug/Viewing Server Dump (DVS) – The DVS dump has two parts:
 - The same diagnostic information available in the Mini Dump, but accessed through the DVS menu system.
 - An IBM system dump that is generated dynamically when the abend occurs. Access this dump only when you need to look at actual system dump storage. StarTool DA/DVS provides you with formatted access to this system dump.

Since this dump is in IBM system dump format, it can be sent to any vendor for additional abend diagnostics.

See [Chapter 2, "Debugging and Viewing Server"](#) on page 17.

Component Overview

StarTool DA consists of the following components:

- [Error Detector Server \(EDS\)](#)
- [Workload Server](#)
- [Debug/Viewing Server \(DVS\)](#)
- [Language Server](#)
- [ChangeMan ZMF Option](#)

The following options are separately licensed:

- [DB2 Option](#)
- [IMS Option](#)

Error Detector Server

The Error Detector Server (EDS) detects any abend condition, intercepts the abend and dump, supplements the dump management capabilities of IBM dump processors, and captures information that would otherwise be unavailable.

The Error Detector logs information to the console or SMF for management and problem reporting. Diagnostic information is written to a Mini Dump Debug Report for you to view in your job output, or formatted for viewing using the Debug/Viewing Server (DVS).

The Error Detector Server resides within z/OS. It is reentrant and provides a maximum level of integrity and a minimum amount of exposure to your production systems.

Workload Server

The Workload Server works behind the scenes. It suppresses duplicate dumps, processes all the captured information from the Error Detector Server, and stores the information in a database structure.

Debug/Viewing Server

The Debug/Viewing Server (DVS) allows you to see your Mini Dump and diagnostic data through an ISPF-based menu system. Although similar to the Mini Dump report, it allows you to choose only the panels you need to diagnose and solve the abend, including those in a shared DASD environment (for remote viewing).

The debug panels are specific to the type of abend, the language of the failing program, and the types of files or databases used.

Your security profile determines the dumps you can access. RAChecks are performed and only those dumps that you can view display in DVS.

Language-specific information shows the abending statement (COBOL, ASM-H, HLASM) as well as the contents of instruction operands. You can view the dump, source code, data division, contents of any field, as well as a language-specific (COBOL, BAL) Debug/Analysis Section.

DVS supports all system dumps that are generated under z/OS. DVS tracks all abend activity, isolates user-defined output, and provides a comprehensive security system. The DVS database index summarizes abend activity in a concise, easy to use format.

Other features of the Debug/Viewing Server (DVS) include:

- COBOL Debug Facilities – allow you to view the dump, source code, data division, and content of fields. The Point-and-shoot facility allows you to trace the COBOL Procedure Division program flow. A pop-up panel shows you the COBOL statement operands and their contents at abend time.
- Multiple Viewing Options – give you more than one way to view your diagnostic data. You can view your output using compatibility mode, ISPF Debug/Viewing Server or IPCS Viewer. Compatibility mode allows you to see your Mini Dump and diagnostic data. Debug/Viewing Server mode is a full ISPF viewer/server. IPCS mode allows you to view system dump information as well as original data using IBM IPCS.
- Remote Viewing – allows you to view dumps from another MVS system. The DBI (database index) must reside on shared DASD.
- Disassembler Function – creates the assembler code from a load module to assist in the dump solving process.
- Formatted Control Blocks – control blocks are formatted into an easily understandable format. DCBs and ACBs are listed and corresponding data items clearly display.
- Last Record Read/Written – the last record read/written for every open file is listed. All access methods are supported (BSAM, QSAM, VSAM, BPAM, and so on).
- Online HELP facilities – contain information on abend codes, reducing the need for abend code manuals.

Language Server

The Language Server analyzes the abend at the source language level. The Language Server allows input source-level debugging of a program, compiler output source-level,

and assembler level debugging. The language database is built at translate, compile, and link-edit time.

When you need detailed source information for a COBOL abend, the Language Server merges actual source code into the Mini Dump Debug Report and creates debug information to be used by the Debug/Viewing Server. At compile time, the Language Server stores the source listing in a language database. The source listing information is automatically available online through the Debug/Viewing Server (DVS).

A significant benefit is the flexibility of obtaining source support. When a compiled listing is not available at abend time, simply recompile before viewing the debug information with DVS.

The Language Server provides:

- The source code of the statement in error displayed in the Failing Instruction Section. All fields in the abending statement are also shown for data related abends.
- The names and contents of all individual fields displayed in the Working Storage Section.
- The Procedure Division source code displayed in the Program Listing Section with the abending statement indicated.

DB2 Option

The DB2 option supports call-attach programs, TSO-attach programs, and stored procedures.

The DB2 options produces the following sections of diagnostic information:

- SQL Call Summary – contains DB2 subsystem name, plan name, program name, statement number, address of CICS SQL call, Plist and SQLCA interpretation.
- Plan Information – contains the plan name, bind date and time, etc.
- Package Information – contains package information, such as precompile date/time, etc.
- DBRM Information – contains information about the DBRM.
- SQL Statement – contains the formatted last SQL statement.
- Host Variables – contains names and contents of the host variables associated with the last SQL call.

IMS Option

- Produces diagnostic information to assist in the analysis of IMS application program dumps.
- Provides information for all PCBs used by the program, and the facility to display the PCB storage in the dump (using the Point-and-Shoot feature in DVS).
- Displays the last DL/I call information, including user-specified parameters, function code, and their addresses.
- Supports region types DL/I Batch, BMP, MPP, IFP (Fast Path), and DBB.
- Supports PCB types TP (I/O and Alternate), DB, Fast Path (MSDB & DEDB), GSAM.
- Supports all PSBGEN languages and most DL/I-Call languages.

ChangeMan ZMF Option

The ChangeMan ZMF Option lets you access information for the package in which the abending program resides. The ZMF option:

- Provides information for the load component and the package containing the abending program.
- Displays history information for the component. All of the packages containing the component appear.
- Displays the ChangeMan ZMF listing.

Chapter 2

Debugging and Viewing Server

Use the Debugging and Viewing Server (DVS) to:

- Delete source listings from the language database
- Delete dump data sets
- Process dump data sets with IPCS
- View StarTool DA dump reports with DVS
- View StarTool DA Batch MVS remote dumps
- View applicable ChangeMan data and program listing

See the following topics for more information:

Starting a DVS Session	18
Using the DVS Panels	22
Multilevel Storage Viewing	52
Program Call/Trace Summary for Inactive Modules	54

Starting a DVS Session

To use the DVS panels, select StarTool DA from your ISPF/PDF menu. If it is not on the ISPF menu, contact your installer for instructions to gain access to StarTool DA.

If StarTool DA CICS and DA are installed at your site, the first panel that displays is the StarTool DA Primary Option Panel allowing you to select either DA CICS or DA Batch.

```

ESPYAP01      ----- StarTool DA Primary Option Panel -----
Command/Option ===>                                     Use PF3/END

Select a StarTool DA application

Your selection ===> 2    1 = StarTool DA CICS
                        2 = StarTool DA Batch
  
```

This example is for DA Batch. The DBI Selection Panel displays from which you select a local or remote DBI (assuming you specified remote DBIs in your WLS JCL).

```

----- StarTool DA : DBI Selection Panel -----
COMMAND ===>

Select local or remote DBI
Local DBI:  _ WST001.DA.E501.E390.DBKF

Remote DBI:ID      Dataset Name
_ Q002            WST001.DA2.E501.E390.DBKF
  
```

When you select a DBI, all processing for the remainder of the session uses the selected DBI. If you want to process a different DBI, you must back out to the DBI Selection Panel and select a different DBI.

When you select the DBI, the Initial DVS Dump Selection Panel displays. If your site has not implemented the StarTool DA remote viewing option, this panel automatically displays first.

```

----- StarTool DA : Initial DVS Dump Selection Panel -----
COMMAND ===>

Overtime any or all selection criteria to view specific
StarTool DA DVS Data Sets.
- O R -
Press enter to view DVS Data Sets with the designated prefix.

DSN Selection Prefix: USER298

Jobname:                Abend Code:                Program Name:

Search Start Date: MM/DD/YYYY

View Source Listing Directory: _

Current Source Listing Directory: WST001.DA.E501.E390.DBKF
  
```

From the Initial DVS Dump Selection Panel, you can select the dumps to view by specifying:

- DSN Selection Prefix

- Jobname
- Abend Code
- Program Name
- Search Start Date
- DDname (View Source Listing Directory)

The default high-level qualifier for the DSN Selection Prefix is the owner ID of the job. The StarTool DA administrator can change this high-level qualifier by modifying the EMCT table.

The Current Source Listing Directory for the language database displays at the bottom of the screen. You can change to a different Source Listing Directory (if you use multiple language databases).

When Exit (PF3) is pressed on the **Initial DVS Dump Selection Panel** (ESPYIPA0), DVS redisplay the **DBI Selection Panel** (ESPYIPAH), allowing the specification of an alternate DBI for processing.

Selecting a Dump Data Set

When you press Enter on the Initial DVS Dump Selection Panel (and the View Source Listing Directory field is blank), the Dump Data Set Selection panel appears. Only those dumps that you are authorized to view display (RACF security checks are performed on your user ID). The dumps listed are selected from those in the selected DBI (either the local DBI or the remote DBI).



NOTE To add remote dumps to your DVS, you must first know where your DBI (Database index) files reside. The remote DBIs are identified by the RDBIxxxx DD statements in the DA WLS (Workload Server) JCL. The xxxx value in the DDname is user-definable. See *Appendix C: Workload Server* in the *Serena® StarTool® DA Installation Guide*.

```

----- StarTool DA : Dump Data Set Selection -----
COMMAND ==>>>                                SCROLL ==>> PAGE

-- Enter 'V' in Selection Field to enter 'Dump View Server' Main Menu.
-- Enter 'I' in Selection Field to Process DVS Data Set with IPCS.
-- Enter 'D' in Selection Field to delete the DUMP data set.
-- Press PF10 or PF11 to display additional selection data.

S Jobname          Program          Abend          Date              Time
- T1407400         PAYROLL7         S0C7 -        DECEMBER 17, 2003 13:29:29
- T1407401         PAYROLL7         S0C7 -        DECEMBER 17, 2003 13:35:00
- T1712360         PAYROLL7         S0C7 -        DECEMBER 17, 2003 13:12:36
- T171236M         PAYROLL7         S0C7 -        DECEMBER 17, 2003 13:07:33
***** Bottom of data *****

```

Selecting Options from the DVS Selection Menu

Type V in the Selection Field to display the DVS Selection Menu screen, shown below. (To view remote dumps, you must have included RDBIxxxx DD statements in the Workload

Server startup PROC to list remote dumps, and you must have selected a remote DBI for dump selection. See *Activating the Workload Server* in the *DA Installation Guide*.)

DVS obtains temporary shared ownership of the dump selected for viewing. Select an option from this menu to diagnose the Abend description in the Title field. The DVS panels are described in [Using the DVS Panels](#).

```

----- StarTool DA : DVS Selection Menu -----COMMAND
====>

Title: ABEND S0C7  IN JOB DA54B9DM AT 11:11:02 ON WEDNESDAY, FEBRUARY 19, 2007

          1  System and Job Information
          2  Diagnostic Section
          3  Failing Instruction Section
          4  Reg Contents and PSW Info
          5  Pgm Call Trace Summary
          6  Program Storage
          7  Pgm Listing (COBOL Only)
          8  Data Mgmt Control Blocks
          C  ChangeMan ZMF Information
          D  Program Call/Trace Details
          L  Load a source listing to DA

          Enter number of selected Option or END(PF3) To EXIT.

```

Processing a Dump Data Set with IPCS

IPCS allows you to view SNAPs as well as SDUMPs using the IBM IPCS. SDUMPs and SNAPs are formatted by DA into IBM Sys1.Dump format. Because all the record types and internal formatting options are the same as IBM SDUMPs, you can use IPCS to view or format output.

To invoke IPCS for a system dump, type I next to the Program name on the Dump Data Set Selection panel. The IPCS Primary Option Menu displays.

```

----- IPCS PRIMARY OPTION MENU -----
OPTION  ==> 0

          0  DEFAULTS      - Specify default dump and options
          1  BROWSE        - Browse dump data set
          2  ANALYSIS      - Analyze dump contents
          3  UTILITY       - Perform utility functions
          4  INVENTORY     - Inventory of problem data
          5  SUBMIT        - Submit problem analysis job to batch
          6  COMMAND       - Enter subcommand, CLIST or REXX exec
          T  TUTORIAL      - Learn how to use the IPCS dialog
          X  EXIT          - Terminate using log and list defaults

          *****
          * USERID - USER298
          * DATE   - 03/15/03
          * JULIAN - 02.196
          * TIME   - 20:06
          * PREFIX - USER298
          * TERMINAL- 3278
          * PF KEYS - 12
          *****

          Enter END command to terminate IPCS dialog

```

To see the IPCS Default Values, choose Option 0 from the IPCS Primary Option Menu; the IPCS Default Values panel displays. The default dump data set name in the Source field is the dump you selected on the Dump Data Set Selection panel. Because StarTool DA formats even SNAP dumps similar to SDUMPs (just smaller), you can use IPCS to do your

viewing and debugging. However, most application dumps are easier to solve using the Debug/Viewing Server (DVS).

```
----- IPCS Default Values -----
Command ==>

You may change any of the defaults listed below. The defaults shown before
any changes are LOCAL. Change scope to GLOBAL to display global defaults.

Scope ==> LOCAL (LOCAL, GLOBAL, or BOTH)

If you change the Source default, IPCS will display the current default
Address Space for the new source and will ignore any data entered in
the Address Space field.

Source ==> DSNAME('USER298.ESPY.D011227.T150307.S001')
Address Space ==>
Message Routing ==> NOPRINT TERMINAL
Message Control ==> CONFIRM VERIFY FLAG(WARNING)
Display Content ==> NOMACHINE REMARK REQUEST NOSTORAGE SYMBOL

Press ENTER to update defaults.

Use the END command to exit without an update.
```

Deleting a DUMP Data Set

When you type D (delete the DUMP data set) in the Selection Field, the system displays a pop-up screen requesting that you confirm the decision to delete the dump data set. If you are sure you want this data set deleted, press the Enter key; otherwise, press the PF3 key to return to the Dump Data Set Selection screen.

```
----- StarTool DA : Dump Data Set Selection -----
COMMAND ==>                                SCROLL ==> PAGE

-- Enter 'V' in Selection Field to enter 'Dump View Server' Main Menu.
-- Enter 'I' in Selection Field to Process DVS Data Set with IPCS.
-- Enter 'D' in Selection Field to delete the DUMP data set.
-- Press *****

e          StarTool DA : Dump Data Set Deletion          e
S Jobname e COMMAND ==>                                  e me
D DA54B9DM e                                             e 1:02
***** e / Confirm request to delete a dump data set.   e *****
e                                                     e
e DSNAME= SRADHAN.ESPY.D030219.T111102.S001           e
e                                                     e
e Press the ENTER key to delete the selected data set. e
e Press PF3 to cancel the delete request.              e
e                                                     e
*****
```

If you delete a data set, the entry still displays, but its status shows that it was deleted.



NOTE You cannot delete a dump that resides on another MVS system. Dumps selected from a remote DBI can be selected for viewing only. You can only delete a dump from the DVS that “owns” the local DBI that contains the dump.

Deleting a Source Listing from the Language Database

If you type Y in the View Source Listing Directory field on the Initial DVS Dump Selection Panel, the Source Module Display panel appears where you can view a directory of source listings within the Language Database. To delete an entry from the directory (DBKF), and from the VSAM source listing file, type D next to the entry.

```

----- StarTool DA : Source Module Display -----
COMMAND ===>                                SCROLL ===> PAGE

Source Listing Directory: WST001.DA.ES01.E390.DBKF

  -- Enter 'D' in Selection Field to delete the Source listing. --

_ Modname: ESPUSACT  Date: 03/15/2007 Time: 07:42:30 Retpd: 0045 Maxgen: 0000
Type: COBOL/MVS          DSN: WST001.DA.ES01.DBDF.D011227.T074230.A018
-----
_ Modname: ESPUSACP  Date: 03/15/2007 Time: 07:42:13 Retpd: 0045 Maxgen: 0000
Type: COBOL/MVS          DSN: WST001.DA.ES01.DBDF.D011227.T074213.A018
-----
_ Modname: ESPUSACM  Date: 03/15/2007 Time: 07:41:58 Retpd: 0045 Maxgen: 0000
Type: COBOL/MVS          DSN: WST001.DA.ES01.DBDF.D011227.T074158.A018
-----
_ Modname: ESPUSACC  Date: 03/15/2007 Time: 07:41:30 Retpd: 0045 Maxgen: 0000
Type: COBOL/MVS          DSN: WST001.DA.ES01.DBDF.D011227.T074130.A018
-----
_ Modname: ESPYCB39  Date: 03/15/2007 Time: 07:38:25 Retpd: 0045 Maxgen: 0000
Type: COBOL/MVS          DSN: WST001.DA.ES01.DBDF.D011227.T073825.A018
-----

```

Using the DVS Panels

The Debug/Viewing Server (DVS) displays your Mini Dump and diagnostic data. Although similar to the Mini Dump Debug Report, it allows you to choose only the panels you need to diagnose and solve the abend. The debug panels are specific to the type of abend, the language of the failing program, and the types of files or databases in use. The entire

COBOL source is viewable if you compiled your program (either before or after the abend) using the DA COBOL Language Server.

```

----- StarTool DA : DVS Selection Menu -----
COMMAND ==>>>

Title: ABEND S0C7  IN JOB USER298A AT 15:03:07 ON SATURDAY, MARCH 15, 2007

          1 System and Job Information
          2 Diagnostic Section
          3 Failing Instruction Section
          4 Reg Contents and PSW Info
          5 Pgm Call Trace Summary
          6 Program Storage
          7 Pgm Listing (COBOL Only)
          8 Data Mgmt Control Blocks
          C ChangeMan ZMF Information
          D Program Call/Trace Details
          L Load a source listing to DA

          Enter number of selected Option or END(PF3) To EXIT.

```

DVS allows you to see all the address space of the abended task without preloading the detail dump data set. For most COBOL programs, it is not necessary or desirable to incur this overhead because only Option 6 (Program Storage) requires this information. You can debug most COBOL programs by using the other menu options.

Assembler and complicated COBOL programs may require looking at all of the address space storage or just the abending program storage. If you select Option 6, you may notice a small delay while the system loads the dump into storage.

DVS is ISPF-based and is a full ISPF viewer/server. The point-and-shoot feature lets you trace through dump control blocks, addresses, and data sections of the program and dump.

To use point-and-shoot, place the cursor on the first or second byte of a hexadecimal address and press Enter. When the cursor is placed on the first byte, you are positioned at the 31-bit address; when the cursor is placed on the second byte, you are positioned at the 24-bit address.



NOTE DVS extracts source statements for viewing as follows:

When DA Batch processes the abend of a batch program, it locates and extracts from the DBKF the source for the abending statement and any associated data elements. It saves those source statements as part of the dump diagnostic data in the DBI data set.

If DA Batch is able to locate the source at the time of the abend, when DVS options 3 and 4 (Failing Instruction Section and Reg Contents and PSW Info) are subsequently selected, the displayed source statements are NOT extracted from the DBKF, they are extracted from the DBI data set.

If DA Batch is unable to locate the source at the time of the abend, DVS will attempt to extract the source from the DBKF when options 3 and 4 are selected.

Option 1 – System and Job Information

System and Job Information is the second panel of the Debug/Viewing Server. This is a standard section that always provides the same type of information. The System and Job Information Panel identifies:

- Date of the abend/error
- StarTool DA version
- Job name
- Step name
- Owner of job
- Time of the error
- Completion code

Other helpful information, in case you need to call Customer Support includes:

- CPU model number
- The operating system release level
- CP FMID
- System on which your program was executing
- Subsystem Release Info (for example: CICS/IMS/DB2)
- DFSMS release level
- JES2 release level

```

----- StarTool DA : System and Job Information -----
COMMAND ===>

Abend Date: March 15, 2007                                DVS Version:
Jobname:   USER298A   CPU Model: 2066   Operating System: MVS/SP 7.0.4
Stepname:  S010      CP FMID:  HBB7707
Owner:     USER298   System:      D001
                                     DFSMS:   V3.3.2           Time: 15:03:07
                                     JES2:    Z/OS 1.4

Completion Code: S0C7-

```

Option 2 – Diagnostic Section

The Diagnostic Section provides information to solve the problem. The cause of the error and corrective actions are described. The information varies, depending upon the programming language used and whether or not the error was external, data related, or something else.

For external errors, the diagnosis includes:

- Cause of the error
- DDNAME and data set name (whenever possible)

See [S813 \(Diagnostic Section Only\)](#) for an example of an external error.

For data-related errors, the Diagnostic Section provides:

- Type of error
- Fields in error
- Location (displacement) of the fields within their respective base locator cell number
- Contents of the fields in error
- Description of the error

Language specific information shows the abending statement (COBOL, ASM-H, HLASM) as well as the contents of instruction operands. It shows the failing COBOL statement and its operand contents if you compiled your program using the StarTool DA COBOL Language Server or used the Repository Data Set (RDS) to store your COBOL compile output.

The following panels show the diagnostic information for an S0C7 abend.

```

----- StarTool DA : Diagnostic Display -----
COMMAND ==>                                SCROLL ==> PAGE

DVS Dump DSN: USER298.ESPY.D011227.T150307.S001
Title: ABEND S0C7 IN JOB USER298A AT 15:03:07 ON SATURDAY, March 15, 2007
Diagnostic Information
SYSTEM ABEND: S0C7

Description: This ABEND is a data exception and can only occur when
             decimal (packed) instructions are used. One of the
             following can cause this error:

             1) The sign or digit codes of one or more bytes
                manipulated by the packed or CONVERT TO BINARY
                instructions is invalid for packed decimal use.
                Packed decimal digits must be in the range 0 through
                9, with only the sign digit being a digit in the range
                A through F.

             2) Fields in decimal (packed) arithmetic overlap
                incorrectly.

             3) A packed decimal multiplicand has too many high-order
                significant digits.

```

```
----- StarTool DA : Diagnostic Display -----  
COMMAND ==>                                SCROLL ==> PAGE
```

```
DVS Dump DSN: USER298.ESPY.D011227.T150307.S001  
Title: ABEND S0C7 IN JOB USER298A AT 15:03:07 ON SATURDAY, March 15, 2007  
Diagnostic Information
```

Possible causes for this abend include:

```
Subscript error, referenced beyond table  
COBOL: working storage not initialized  
      bad data, should check data for errors  
      garbage in a field being tested or displayed  
      move zeroes to group level is display, had sublevels  
        that were not  
      period missing after imperative statements  
      within AT END clause  
      binary field in an arithmetic operation is not large  
        enough to accept result
```

```
User Action: Correct the format of the data being manipulated by the  
             packed decimal instructions in the program and rerun the  
             job.
```

Option 3 – Failing Instruction Section

The Failing Instruction section provides the statement in error. Additional information includes:

- Failing Program name and offset
- Instruction sequence prior to and including abending instruction
- Compile date of the program compile date and Program-ID of compiler/assembler
- Load module name and load library name
- Linkage Editor Date of the abending program.

The information provided depends on the programming language used. The following examples show an Assembler program and a COBOL-LE program.

Assembler program example

```

----- StarTool DA : Failing Instruction Information -----
COMMAND ==>

The Failing Instruction is in Program (CSECT):

        ESPYAT39 at offset X' 00046E '

The Failing Instruction is:

==> AP   X'478'(4,R10),X'47C'(7,R10)   <===
        ***** Failing Instruction Analysis *****

        Instruction Type= SS  Number of Operands= 2

        OP1 Type= ADDR ADDR= 00006478 Data= 0000001C
ER> OP2 Type= ADDR ADDR= 0000647C Data= C995A581 938984

```

COBOL-LE program example



NOTE If you compile the program in ChangeMan ZMF, it must use the LIST and MAP options to see the failing COBOL statement.

```

----- StarTool DA : Failing Statement Information -----
COMMAND ==>

The Failing Instruction is in Program (CSECT):

        PAYROLL7 at offset X'0003DA'

The Failing COBOL Statement is:

==> 000436      ADD S0C7-BADFIELD TO S0C7-GOODFIELD
==> 05 S0C7-GOODFIELD          PIC S9(5) COMP-3 VALUE 0.
        Value: 00000C
==> 05 S0C7-BADFIELD REDEFINES S0C7-BADFIELD PIC S9(5) COMP-3.
        Value: 828184                                <-- BAD SIGN DIGIT

```

For SOCx abends, the failing COBOL statement and its operand content display. Normally, this is all you need to solve and correct the problem. However, if you need to see more of the program's source to solve the problem, tab to the failing COBOL statement and point-and-shoot on this statement. The Debug/Viewing Server takes you to Section 7, Pgm Listing Section, and positions the panel on the failing statement. Press PF7 (UP) to examine the COBOL source prior to the failing instruction.

Option 4 – Register and PSW Information

The Register and Program Status Word (PSW) Information section displays Supporting Environmental Data, such as:

- StarTool DA Dump Title, including time and date of abend
- Job name
- Step name

- Program name
- Abend code
- Offset within load model of error
- Instruction length of failing instruction
- Abending Program PSW (Program Status Word)
- Failing Instruction that precipitated the abend
- Register contents (hex) at time of error and register descriptions
- Load module DDname, load module name, and the load library name
- Dump/Viewer Server Data Set Name that contains the raw dump

```

----- StarTool DA : Regs and PSW Display -----
COMMAND ==>

Title: ABEND S0C7  IN JOB USER298A AT 15:03:07 ON SATURDAY, March 15, 2007

Jobname: USER298A  Stepname: S010      Program Name: ESPYCB39
Abend: S0C7-07    Offset: 000D3A

ILC: 06              INTC: 07              PSW: 078D3000  80007710

Failing Instruction: AP  X'138'(3,R8),X'13B'(3,R8)
000436      ADD S0C7-BADFIELD TO S0C7-GOODFIELD

REGISTERS:
0-7  0002B1C8 000070DE 0001C7FC 000080A6 000086D8 80008062 00018808 00050060
AR:  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
8-F  00008998 00008140 00006B00 00007414 00006ACC 0002B018 8000770A 8D8F6B98
AR:  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

Program Loaded From: DDNAME: STEPLIB  LOADMOD: ESPYCB39
DVS Dump Dsname: USER298.ESPY.D011227.T150307.S001
Compile Time: 07:38:25
Compile Date: 27 DEC 2002  LKED Date: 27 DEC 2002  Pgm Type: COBOL/MVS

```

For SOCx abends, the failing COBOL statement and its operand content display. This is usually all you need to solve and correct the problem.

However, if you need to see more of the program source to solve the problem, tab twice to the failing COBOL statement and point-and-shoot on this statement. The Debug/Viewing Server takes you to Section 7, Pgm Listing Section, and positions the panel on the failing statement. Press PF7 (UP) to examine the COBOL source prior to the failing instruction.

```

----- StarTool DA : COBOL Listing -----
COMMAND ==>                                SCROLL ==> PAGE

Selected Job= USER298A
Program=     ESPYCB39

ER STMT> 000436      add S0C7-BadField to S0C7-GoodField
000437      .
000438*-----*
000439 P03-PerformS806 section.
000440*-----*
000441      display MyProgramId ',P03,031I,S806 abend requested.'
000442      call S806-PgmName
000443      .
000444*-----*
000445 P04-PerformMult section.
000446*-----*
000447      display MyProgramId ',P04,041I,Multi-csect abend requested.'
000448      call 'espycb3a' using BadP04Data, S0C7-GoodField
000449      .
000450*-----*
000451 P10-ProcessJclParms section.
000452*-----*

```

If you press the tab key once from the first panel (Regs and PSW Display), the system positions you on the PSW information. If you then press Enter, the system takes you to one instruction beyond the abending instruction (except for certain OC4s). To obtain the abending instruction, take note of the ILC (Instruction Length Count) and subtract this amount from the PSW address.



NOTE Point-and-shoot resets the top of table for ISPF so that when you press PF7 (UP), the system returns to this position. You can also put the cursor on any register and point-and-shoot to go to Section 6.

```

----- StarTool DA : Storage Display -----
COMMAND ===> -6                                SCROLL ===> PAGE

Selected Job= USER298A
Start Address= 00007710 Area= PVT Program= ESPYCB39

Address  Offset      Data (Hexadecimal)      Data (EBCDIC)
00007710 000D40 F8228138 81385830 D0D407F3 5820905C * 8.a.a....M.3... *
00007720 000D50 58F0202C 4110A5C8 05EFD207 D1988148 * .0....vH..K.Jqa. *
00007730 000D60 DC07D198 A2E0D203 D1A0A47A 4130D198 * ..Jqs\K.J.u:...Jq *
00007740 000D70 5030D1A4 4110D1A0 58F02100 05EF5830 * &.Ju..J..0..... *
00007750 000D80 912440F0 30005840 D0D807F4 5820905C * j. 0... .Q.4... *
00007760 000D90 58F0202C 4110A5B2 05EF4130 813E5030 * .0....v.....a.&. *
00007770 000DA0 D1984130 81385030 D19C9680 D19CD203 * Jq...a.&.J.o.J.K. *
00007780 000DB0 D1A0A476 4130A28C 41409154 4150D198 * J.u...s..j.&Jq *
00007790 000DC0 5030D1A4 5040D1A8 5050D1AC 4110D1A0 * &.Ju& Jy&&J...J. *
000077A0 000DD0 58F02100 05EF5830 912440F0 30005840 * .0.....j. 0... *
000077B0 000DE0 D0DC07F4 D2038020 6002D503 8020A2CD * ...4K...-.N...s. *
000077C0 000DF0 4780B3E6 D5038020 A1744780 B3E6D503 * ...WN...~....WN. *
000077D0 000E00 8020A155 4780B3E6 D5038020 A2C94780 * ..~....WN...sI... *
000077E0 000E10 B3E658F0 202C4110 A59C05EF 58F0202C * .W.0....v....0... *
000077F0 000E20 4110A586 05EF92F2 81505830 D0B407F3 * ..vf...k2a&....3 *
00007800 000E30 D2039170 A0A05820 91701A28 D2018158 * K.j....j...K.a. *
00007810 000E40 2720D203 9170A09C D2039174 A0A05820 * ..K.j...K.j..... *

```

If you subtract the abending instruction length (ILC) of 6, by typing -6 on the command line, the following panel (Storage Display) aligns you on the abending instruction. This example is used only to illustrate the point-and-shoot facility; the Regs and PSW Display panel shows the actual abending instruction.

```

----- StarTool DA : Storage Display -----
COMMAND ===>                                SCROLL ===> PAGE

Selected Job= USER298A
Start Address= 0000770A Area= PVT Program= ESPYCB39

Address  Offset      Data (Hexadecimal)      Data (EBCDIC)
0000770A 000D3A FA228138 813BF822 81388138 5830D0D4 * .a.a.8.a.a....M *
0000771A 000D4A 07F35820 905C58F0 202C4110 A5C805EF * .3...*.0....vH.. *
0000772A 000D5A D207D198 8148DC07 D198A2E0 D203D1A0 * K.Jqa...Jqs\K.J. *
0000773A 000D6A A47A4130 D1985030 D1A44110 D1A058F0 * u:...Jq&.Ju..J..0 *
0000774A 000D7A 210005EF 58309124 40F03000 5840D0D8 * .....j. 0... .Q *
0000775A 000D8A 07F45820 905C58F0 202C4110 A5B205EF * .4...*.0....v... *
0000776A 000D9A 4130813E 5030D198 41308138 5030D19C * .a.a.&.Jq..a.&.J. *
0000777A 000DAA 9680D19C D203D1A0 A4764130 A28C4140 * o.J.K.J.u...s... *
0000778A 000DBA 91544150 D1985030 D1A45040 D1A85050 * j..&Jq&.Ju& Jy&& *
0000779A 000DCA D1AC4110 D1A058F0 210005EF 58309124 * J...J..0.....j. *
000077AA 000DDA 40F03000 5840D0DC 07F4D203 80206002 * 0... ..4K...-.. *
000077BA 000DEA D5038020 A2CD4780 B3E6D503 8020A174 * N...s....WN...~. *
000077CA 000DFA 4780B3E6 D5038020 A1554780 B3E6D503 * ...WN...~....WN. *
000077DA 000E0A 8020A2C9 4780B3E6 58F0202C 4110A59C * .sI...W.0....v. *
000077EA 000E1A 05EF58F0 202C4110 A58605EF 92F28150 * ...0....vf...k2a& *
000077FA 000E2A 5830D0B4 07F3D203 9170A0A0 58209170 * .....3K.j....j. *
0000780A 000E3A 1A28D201 81582720 D2039170 A09CD203 * ..K.a...K.j...K. *

```

In summary, if you point-and-shoot on any address (PSW or registers), the system takes you to Section 6: Pgm Storage to view the hexadecimal listing of storage. If you point-and-shoot on the failing COBOL instruction, the system takes you to Section 7: Pgm Listing Section to view the source code.

From the Regs and PSW Display panel, you can also use DA commands. For example, you can locate an area in storage using a register number, with or without offsets. If you type L R4+90 on the command line, the Storage Display panel displays. This is also useful for navigating control blocks.

```

----- StarTool DA : Regs and PSW Display -----
COMMAND ==> L R4+90

Title: ABEND S0C7  IN JOB USER298A AT 15:03:07 ON SATURDAY, March 15, 2007

Jobname: USER298A  Stepname: S010      Program Name: ESPYCB39
Abend: S0C7-07    Offset: 000D3A

ILC: 06              INTC: 07              PSW: 078D3000  80007710

Failing Instruction: AP  X'138'(3,R8),X'13B'(3,R8)
000436      ADD S0C7-BADFIELD TO S0C7-GOODFIELD

REGISTERS:
0-7  0002B1C8 000070DE 0001C7FC 000080A6 000086D8 80008062 00018808 00050060
AR:  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
8-F  00008998 00008140 00006B00 00007414 00006ACC 0002B018 8000770A 8D8F6B98
AR:  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

Program Loaded From: DDNAME: STEPLIB  LOADMOD: ESPYCB39
DVS Dump Dsname: USER298.ESPY.D011227.T150307.S001
Compile Time: 07:38:25
Compile Date: 27 DEC 2002  LKED Date: 27 DEC 2002  Pgm Type: COBOL/MVS

```

This is the resulting Storage Display panel.

```

----- StarTool DA : Storage Display -----
COMMAND ==>                                SCROLL ==> PAGE

Selected Job= USER298A
Start Address= 00008768  Area= PVT  Program= ESPYCB39

Address  Offset      Data (Hexadecimal)      Data (EBCDIC)
00008768 001D98  E5E2C1D4  F0F0F0F2  00000000  00010000  * VSAM0002..... *
00008778 001DA8  00000000  0000736C  0004B7D0  000088E0  * .....%.....h\ *
00008788 001DB8  F0F2F323  04080010  00000000  00000000  * 023..... *
00008798 001DC8  00000000  00000000  23208800  00000000  * .....h..... *
000087A8 001DD8  00000000  00000310  00000000  00000000  * ..... *
000087B8 001DE8  00000000  00000000  00000000  00000000  * ..... *
000087C8 001DF8  00000000  00000000  00000000  00000000  * ..... *
000087D8 001E08  00000000  00000000  00000000  00000000  * ..... *
000087E8 001E18  00000000  00000000  00000000  00000000  * ..... *
000087F8 001E28  00000000  00000000  00000000  00000000  * ..... *
00008808 001E38  00000000  00000000  00000000  00000000  * ..... *
00008818 001E48  00000000  00000000  00000000  00000000  * ..... *
00008828 001E58  00000000  00000000  00000000  00000000  * ..... *
00008838 001E68  00000000  00000000  00000000  00000000  * ..... *
00008848 001E78  00000000  00000000  00000000  00000000  * ..... *
00008858 001E88  00000000  00000000  00000000  00000000  * ..... *
00008868 001E98  00000000  00000000  00000000  00000000  * ..... *

```

Option 5 – Program Call Trace Summary

The Program Call/Trace Summary shows you which programs were called and in what order. The initial panel gives the summary of the call and trace information and enables you to select the called programs for additional detail.

- Selectively display the formatted program storage for all COBOL programs in the calling chain. Included are the Task Global Table (TGT), Dynamic Storage Area (DSA), BLW cells, and BLL cells.
- Display of the source (option S) and trace table (option D) directly from this option rather than returning to the main DVS menu.
 - The display generated for option S is the same as that generated for option 7 on the main DVS menu
 - The display generated for option D is the same display as that generated for option D on the main DVS menu



NOTE This option provides the DVS menu equivalent of the COBOL Data Section in the minidump.

Viewing the Program Call and Trace Summary and Selecting a Called Program

When you select option 5 from the DVS Selection Menu, the Program Call / Trace Summary panel appears with a list of called programs.

```

ESPYIPAA -- ChangeMan ZMF-DA --- Program Call/Trace Summary - Row 1 to 2 of 2
COMMAND ==>>> SCROLL ==>> PAGE

          Call/Trace Summary and Application Program Attributes

To display a called program's COBOL storage and data, enter "S" into the
selection column and press enter. (Press PF1 for additional help).

---Called Pgm--- ----Called Pgm Attributes---- -----Caller----- ---Call-----
S Program Load-Mod Comp Date Length Language Load-Mod Program Offset Mode
- - - - -
ESPYCB39 ESPYCB39 25 Apr 2011 006F8E COBOL/zOS *SYSTEM* Call
_ ESPYCB39 ESPYCB39 25 Apr 2011 006F8E COBOL/zOS CEEBINIT CEEBTOR 000000 Call
***** Bottom of data *****

```

The Call Trace Summary provides the following information:

- Called/linked programs on the save area chain
- Program locations where the calls occurred
- Identification of the program in error, when available
- Application Program Attributes listing:
 - Each load module and its program name
 - Compile Date, Length, and Language for each program (COBOL, HLA Assembler, and so on)

You can view program and data storage for selected called COBOL programs in the calling chain of the abending program.

Only called programs that have completed COBOL initialization processing are eligible for additional displays. DA Batch marks the programs that meet this criteria by placing an underscore in the S column of the called program's row.

Selecting COBOL Program Details to Display

To select one of these called programs for processing, replace the underscore with an S and press enter.

The COBOL Program Detail Selection Menu appears.

```

ESPYIPAI - ChangeMan ZMF-DA -- COBOL Program Detail Selection Menu -----
Command/Option ==>                                     Use PF3/END to exit

COBOL program name : ESPYCB39          Compile time/date : 2011/04/25 at 13:02:46
                                      COBOL compiler      : COBOL/zOS   at 04.01.00

      COBOL program storage display options
      1 Display Linkage-Section (BLL) storage
      2 Display Working-Storage (BLW) storage
      3 Display program storage

      S Display program source listing
      D Display program call/trace details

      COBOL control block display options
      4 Display COBOL TGT
      5 Display COBOL RunCom
      6 Display COBOL DSA

Enter number of selected Option or END(PF3) to exit.

```

Select from the options on the COBOL Program Detail Selection Menu to investigate the cause of the abend. The following options are provided.

- "Displaying BLL Storage" on page 33
- "Displaying BLW Storage" on page 34
- "Displaying Program Storage" on page 34
- "Displaying the Program Listing of the Program" on page 35
- "Displaying the Program Call and Trace Detail" on page 35
- "Displaying the COBOL TGT Control Blocks" on page 35
- "Displaying the COBOL RUNCOM Control Blocks" on page 36
- "Displaying the COBOL DSA Control Blocks" on page 37

Displaying BLL Storage

To display the COBOL Linkage-Section (BLL) storage for the selected called program, enter 1 in the command line.

The COBOL BLL Storage Display Menu appears.

```

ESPYPYPAJ - ChangeMan ZMF-DA - COBOL BLL Storage Display Menu ----- Row 1 of 3
COMMAND ==>>                                SCROLL ==>> PAGE

Current COBOL program : ESPYCB39

  Input  Cell      Address      Length      Description
      1      0          0          0          BLL 1      -address is zero-
      2     16113FD0      6          6          BLL 2
      3      0          0          0          BLL 3      -address is zero-
***** Bottom of data *****
    
```

Displaying BLW Storage

To display the COBOL Working-Storage (BLW) storage for the selected called program, enter 2 in the command line.

The COBOL BLW Storage Display Menu appears.

```

ESPYPYPAJ - ChangeMan ZMF-DA - COBOL BLW Storage Display Menu ----- Row 1 of 10
COMMAND ==>>                                SCROLL ==>> PAGE

Current COBOL program : ESPYCB39

  Input  Cell      Address      Length      Description
      1     163C80B8     1000      1000      BLW 1
      2     163C90B8     1000      1000      BLW 2
      3     163CA0B8     1000      1000      BLW 3
      4     163CB0B8     1000      1000      BLW 4
      5     163CC0B8     1000      1000      BLW 5
      6     163CD0B8     1000      1000      BLW 6
      7     163CE0B8     1000      1000      BLW 7
      8     163CF0B8     1000      1000      BLW 8
      9     163D00B8     1000      1000      BLW 9
     10     163D10B8     1000      1000      BLW 10      -length forced-
***** Bottom of data *****
    
```

Displaying Program Storage

To display the program storage details for the selected called program, enter 3 in the command line.

The COBOL Program Detail Storage Display panel appears.

```

ESPYIPB1 -- ChangeMan ZMF-DA - COBOL Program Detail Storage Display Row 1 of 256
COMMAND ==>> SCROLL ==>> PAGE

Selected Job= BGH#9DM2 Display: 1 of 1
Start Address= 16100440 Area= EPVT Program= ESPYCB39 PROGRAM

Address Offset Data (Hexadecimal) Data (EBCDIC)
16100440 0 47F0F028 00C3C5C5 00000368 00000014 * .00..CEE..... *
16100450 10 47F0F001 98CEAC00 161004F6 00000000 * .00.q.....6.... *
16100460 20 00000000 00000000 90ECD00C 4110F038 * .....0. *
16100470 30 98EFF04C 07FF0000 16100440 00000000 * q.0<..... *
16100480 40 16106588 161004EE 16100440 161023EC * ...h..... *
16100490 50 161074C0 1610050A 00104001 00000008 * ..... *
161004A0 60 C5E2D7E8 C3C2F3F9 F2F0F1F1 F0F4F2F5 * ESPYCB3920110425 *
161004B0 70 F1F3F0F2 F4F6F0F4 F0F1F0F0 04740000 * 130246040100.... *
161004C0 80 1400076C E4E86C4C 20A1BC80 508C410F * ...%UY%<.~.&... *
161004D0 90 810844D0 0C000000 80C98400 00000208 * a.....Id..... *
161004E0 A0 0000019A 20008000 40404040 0008C5E2 * ..... ..ES *
161004F0 B0 D7E8C3C2 F3F90500 00011610 74100000 * PYCB39..... *
16100500 C0 0000FFFF FFB21610 04401610 04780000 * ..... *
16100510 D0 00080000 00061610 04781610 74101610 * ..... *
16100520 E0 73E80000 00050000 00000000 00000000 * .Y..... *

```

Displaying the Program Listing of the Program

To display the COBOL source listing for the program, enter S in the command line. The COBOL Listing panel appears.

This option displays the same information that is displayed from option 7 on the main DVS menu.

Displaying the Program Call and Trace Detail

To display the program call and trace detail for the abending program, enter D in the command line.

The COBOL Call/Trace Detail panel appears.

```

ESPYIPDD -- ChangeMan ZMF-DA --- COBOL Call/Trace Detail ---- Row 1 to 2 of 2
Command ==>> SCROLL ==>> PAGE

DVS Dump DSN: BHITEFI.ESPY.D110510.T042129.S001
Title: Abend U0929 Job:BGH#IMS2 at 04:21:29 on Tuesday , May 10, 2011

***** Program Call/Trace Details *****
Csect/lmod *SYSTEM*/*SYSTEM* calls csect/lmod DFSDLB00/DFSDBL0 at +X" ".
Csect/lmod COMMON /DFSDBL0 calls csect/lmod DFSDBLM0/DFSDBLM0 at +X"0FA2".
***** Bottom of data *****

```

This option displays the same information that is displayed from option D on the main DVS menu.

Displaying the COBOL TGT Control Blocks

To display the COBOL TGT control block for the abending program, enter 4 in the command line.

The COBOL Program Detail Storage Display panel appears. TGT appears after the program name in the upper right of the panel to identify the type of control blocks displayed.

```

ESPYIPB1 -- ChangeMan ZMF-DA --- COBOL Program Detail Storage Display - Row 1 of 33
COMMAND ==>> SCROLL ==>> PAGE

Selected Job= BGH#9DM2 Display: 1 of 1
Start Address= 1639B448 Area= EPVT Program= ESPYCB39 TGT

Address Offset Data (Hexadecimal) Data (EBCDIC)
1639B448 0 00000000 00000000 00000000 00000000 * ..... *
1639B458 10 00000000 00000000 00000000 00000000 * ..... *
1639B468 20 00000000 00000000 00000000 00000000 * ..... *
1639B478 30 00000000 00000000 00000000 00000000 * ..... *
1639B488 40 00000000 00000000 F3E3C7E3 00000000 * .....3TGT.... *
1639B498 50 06000000 66020260 1639B100 000127FC * .....-..... *
1639B4A8 60 1639B658 00000007 00009CD8 00000000 * .....Q.... *
1639B4B8 70 00000000 000172E0 00000000 00000000 * .....\..... *
1639B4C8 80 16115B58 00000210 00000000 00000000 * ..$...... *
1639B4D8 90 00000000 00000001 E2E8E2D6 E4E34040 * .....SYSOUT *
1639B4E8 A0 C9C7E9E2 D9E3C3C4 00000000 00000000 * IGZSR TCD..... *
1639B4F8 B0 00000000 00000000 00000000 00000000 * ..... *
1639B508 C0 00000000 00000000 00000000 00000000 * ..... *
1639B518 D0 00000000 00000000 00000000 00000000 * ..... *
1639B528 E0 00000000 00000000 1610053C 00000000 * ..... *

```

Displaying the COBOL RUNCOM Control Blocks

To display the COBOL RUNCOM control block for the abending program, enter 5 in the command line.

The COBOL Program Detail Storage Display panel appears. RUNCOM appears after the program name in the upper right of the panel to identify the type of control blocks displayed.

```

ESPYIPB1 -- ChangeMan ZMF-DA --- COBOL Program Detail Storage Display - Row 1 of 64
COMMAND ==>> SCROLL ==>> PAGE

Selected Job= BGH#9DM2 Display: 1 of 1
Start Address= 1639B100 Area= EPVT Program= ESPYCB39 RUNCOM

Address Offset Data (Hexadecimal) Data (EBCDIC)
1639B100 0 C3F3D9E4 D5C3D6D4 000002D8 00C60000 * C3RUNCOM...Q.F.. *
1639B110 10 161147C0 00000002 00006F58 00000000 * .....?..... *
1639B120 20 00000000 16100440 1639B3F0 00000000 * .....0.... *
1639B130 30 161148E0 00000000 00000000 00000000 * ..\..... *
1639B140 40 00012A80 000121BC 00000000 000127FC * ..... *
1639B150 50 00000000 00000000 16115B58 00000000 * .....$. *
1639B160 60 00000000 00017CF0 00000000 00000000 * .....@0..... *
1639B170 70 00000000 F0F0F0F0 F0F0F0F0 00000000 * ....00000000.... *
1639B180 80 00000000 00000000 00000000 1639B3F0 * .....0 *
1639B190 90 00000000 00000000 00000000 00000000 * ..... *
1639B1A0 A0 00000000 00000000 0001D4DE 0001E8CC * .....M...Y. *
1639B1B0 B0 0001E91A 0001E952 0001E98A 0001E9EE * ..Z...Z...Z...Z. *
1639B1C0 C0 00000000 00000000 0001EB92 00000000 * .....k.... *
1639B1D0 D0 00000000 00000000 00000000 00000000 * ..... *
1639B1E0 E0 00000000 00000000 00000000 00000000 * ..... *

```

Displaying the COBOL DSA Control Blocks

To display the COBOL DSA control block for the abending program, enter 6 in the command line.

The COBOL Program Detail Storage Display panel appears. DSA appears after the program name in the upper right of the panel to identify the type of control blocks displayed.

```

ESPYIPB1 -- ChangeMan ZMF-DA --- COBOL Program Detail Storage Display -- Row 1 of 6
COMMAND ==>> SCROLL ==>> PAGE

Selected Job= BGH#9DM2                      Display: 1  of 1
Start Address= 1639F030 Area= EPVT           Program= ESPYCB39 DSA

Address  Offset      Data (Hexadecimal)                Data (EBCDIC)
1639F030    0  00104001  16116600  00017D00  96103382  * .. .....'.0..b *
1639F040   10  96324198  1639F398  16101AF6  000127FC  * o..q..3q...6.... *
1639F050   20  161015A0  161049EC  161049EC  00021166  * ..... *
1639F060   30  00000000  163C80B8  1639B448  161005A0  * ..... *
1639F070   40  161032BA  1610053C  00000000  1639F398  * .....3q *
1639F080   50  00000000  00000000  1639F030  1639B448  * .....0..... *
***** Bottom of data *****

```

Option 6 – Program Storage

When you select Option 6, the Storage Display Selection Panel displays, where you can choose the initial area of storage you want to view. This option loads the detail dump data set for full screen viewing. You can see all of the abended task address space, control blocks, and data areas. You may notice a small delay while the dump is loaded into storage.

```

----- StarTool DA : Storage Display Selection Panel -----
COMMAND ==>>

DVS DSN: USER298.ESPY.D011227.T150307.S001

Please select initial area of storage to view.

_ Area of Abending Instruction
S Storage of Entire Abending Program
_ Lowest Address in the Address Space
_ CVT          (System Communication Vector Table)
_ ASCB         (Job Address Space Control Block)
_ TCB          (Failing Task Control Block)
_ COBOL Working Storage (COBOL Programs Only)

```

To select where to start the storage display, type an S in the line command field and press Enter. Most users choose Storage of Entire Abending Program as a place to start.

Following is the Storage Display panel that appears when you choose Storage of Entire Abending Program.

```

----- StarTool DA : Storage Display -----
COMMAND ==>>>                                SCROLL ==>> PAGE

Selected Job= USER298A
Start Address= 000069D0 Area= PVT Program= ESPYCB39

Address  Offset      Data (Hexadecimal)      Data (EBCDIC)
000069D0 000000  47F0F028 00C3C5C5 000001B0 00000014 * .00..CEE..... *
000069E0 000010  47F0F001 98CEAC00 00006A86 00000000 * .00.q.....f... *
000069F0 000020  00000000 00000000 90ECD00C 4110F038 * .....0. *
00006A00 000030  98EFF04C 07FF0000 000069D0 00000000 * q.0<..... *
00006A10 000040  00008140 00006A7E 000069D0 000074A6 * ..a...=.....w *
00006A20 000050  0000A4A8 00006A9A 00104001 00000008 * ..uy..... *
00006A30 000060  C5E2D7E8 C3C2F3F9 F2F0F0F1 F1F2F2F7 * ESPYCB3920011227 *
00006A40 000070  F0F7F3F8 F2F5F0F1 F0F2F0F0 00000000 * 073825010200... *
00006A50 000080  00000000 E4E82C4C 00A19C80 508C010F * ....UY.<..&... *
00006A60 000090  81004000 0C000000 80C10000 000000D5 * a. ....A....N *
00006A70 0000A0  000000AC 00000000 40404040 0008C5E2 * .....ES *
00006A80 0000B0  D7E8C3C2 F3F90500 00010000 A3F80000 * PYCB39.....t8.. *
00006A90 0000C0  0000FFFF FFB20000 69D00000 6A080000 * ..... *
00006AA0 0000D0  00080000 00060000 6A080000 A3F80000 * .....t8.. *
00006AB0 0000E0  A3D00000 00050000 00000000 00000000 * t..... *
00006AC0 0000F0  00000000 00000000 00000000 000186A0 * .....f. *
00006AD0 000100  00002710 3B9ACA00 05F5E100 00000001 * .....5..... *

```

Commands for the Storage Display Panel

Use the following commands on the Storage Display panel.

LOCATE

Locates specific fields within the Storage Display panel. If the STARTING ADDRESS of the requested area is not in the dump, a message displays saying 'Starting Address is not in the dump'. The syntax is:

```
LOCATE operand or L operand
```

The following operands are supported:

addr	Displays the requested hexadecimal address
PSA	Displays Low Storage location 0 (zero)
PVT	Displays the starting address of 'Low Private Area'
CSA	Displays the starting address of 'Low Common Storage Area (CSA)'
LPA	Displays the starting address of 'Low Link Pack Area (LPA)'
MLPA	Displays the starting address of 'Low Modified Link Pack Area (MLPA)'
FLPA	Displays the starting address of 'Low Fixed Link Pack Area (FLPA)'
SQA	Displays the starting address of 'Low System Queue Area (SQA)'
NUC	Displays the starting address of the Nucleus
ESQA	Displays the starting address of 'High System Queue Area (ESQA)'

ELPA	Displays the starting address of 'High Link Pack Area (ELPA)'
EFLPA	Displays the starting address of 'High Fixed Link Pack Area (EFLPA)'
EMLPA	Displays the starting address of 'High Modified Link Pack Area (EMLPA)'
ECSA	Displays the starting address of 'High Common Storage Area (ECSA)'
EPVT	Displays the starting address of 'High Private Area'
PREV	Returns to the address that was displayed prior to the current address

FIND

Finds a string. The FIND command operates across all storage in the address space. The syntax is:

FIND *string* [prev] or F *string* [prev]

where *string* is either a hex or character string.

A hex string is represented as X'hex string' with an even number of bytes. A character string is represented as either C'character string' or just 'character string'.

In the following Storage Display panel, a FIND command is typed on the Command line.

```

----- StarTool DA : Storage Display -----
COMMAND ==> f 'multi-csect'                                SCROLL ==> PAGE

Selected Job= USER298A
Start Address= 000069D0 Area= PVT Program= ESPYCB39

Address  Offset      Data (Hexadecimal)      Data (EBCDIC)
000069D0  000000  47F0F028 00C3C5C5 000001B0 00000014 * .00..CEE..... *
000069E0  000010  47F0F001 98CEAC00 00006A86 00000000 * .00.q.....f... *
000069F0  000020  00000000 00000000 90ECD00C 4110F038 * .....0. *
00006A00  000030  98EFF04C 07FF0000 000069D0 00000000 * q.0<..... *
00006A10  000040  00008140 00006A7E 000069D0 000074A6 * ..a...=. ....w *
00006A20  000050  0000A4A8 00006A9A 00104001 00000008 * ..uy..... *
00006A30  000060  C5E2D7E8 C3C2F3F9 F2F0F0F1 F1F2F2F7 * ESPYCB3920011227 *
00006A40  000070  F0F7F3F8 F2F5F0F1 F0F2F0F0 00000000 * 073825010200... *
00006A50  000080  00000000 E4E82C4C 00A19C80 508C010F * ....UY.<..&... *
00006A60  000090  81004000 0C000000 80C10000 000000D5 * a. ....A....N *
00006A70  0000A0  000000AC 00000000 40404040 0008C5E2 * ..... ..ES *
00006A80  0000B0  D7E8C3C2 F3F90500 00010000 A3F80000 * PYCB39.....t8.. *
00006AA0  0000D0  00080000 00060000 6A080000 A3F80000 * .....t8.. *

```

When you press Enter, the screen is positioned at the character string 'multi-csect'.

```

----- StarTool DA : Storage Display -----
COMMAND ==>>                                SCROLL ==>> PAGE

Selected Job= USER298A
Start Address= 00006BF0 Area= PVT Program= ESPYCB39

Address  Offset      Data (Hexadecimal)      Data (EBCDIC)
00006BF0 000220  D4A493A3 896083A2 8583A340 81828595 * Multi-csect aben *
00006C00 000230  84409985 98A485A2 A385844B 6BD7F1F0 * d requested.,P10 *
00006C10 000240  6BF1F0F1 C56BC995 A5819389 8440D1C3 * ,101E,Invalid JC *
00006C20 000250  D3409781 99819485 A385994B 6BD7F1F0 * L parameter.,P10 *
00006C30 000260  6BF1F0F2 C56BC5A7 8583A4A3 89969540 * ,102E,Execution *
00006C40 000270  A3859994 899581A3 85844B6B D7F0F36B * terminated.,P03, *
00006C50 000280  F0F3F1C9 6BE2F8F0 F6408182 85958440 * 031I,S806 abend *
00006C60 000290  998598A4 85A2A385 844B6BD7 F0F26BF0 * requested.,P02,0 *
00006C70 0002A0  F2F1C96B E2F0C3F7 40818285 95844099 * 21I,S0C7 abend r *
00006C80 0002B0  8598A485 A2A38584 4B6BD7F0 F16BF0F1 * equested.,P01,01 *
00006C90 0002C0  F1C96BE2 F0C3F440 81828595 84409985 * 1I,S0C4 abend re *
00006CA0 0002D0  98A485A2 A385844B 6BD7F0F0 6BF0F0F1 * quested.,P00,001 *
00006CB0 0002E0  C96BD799 968385A2 A2899587 408881A2 * I,Processing has *

```

DAS (or DISASM)

DAS disassembles hexadecimal dump data where the cursor is positioned. If the cursor is not on valid hexadecimal data, the command is ignored. To execute this command, type DAS or DISASM on the command line; then position the cursor on hexadecimal data and press Enter. You can assign DAS or DISASM to a PF key.

In the following Storage Display panel, the DAS command is typed on the command line. Then, the cursor is positioned on a hexadecimal dump location.

```

----- StarTool DA : Storage Display -----
COMMAND ==>> DAS                                SCROLL ==>> PAGE

Selected Job= USER298A
Start Address= 000076B8 Area= PVT Program= ESPYCB39

Address  Offset      Data (Hexadecimal)      Data (EBCDIC)
000076B8 000000  47F0F028 00C3C5C5 000001D8 00000014 * .00..CEE...Q.... *
000076C8 000010  47F0F001 98CEAC00 0000776E 00000000 * .00.q.....>.... *
000076D8 000020  00000000 00000000 90ECD00C 4110F038 * .....0. *
000076E8 000030  98EFF04C 07FF0000 000076B8 00000000 * q.0<..... *
000076F8 000040  00009158 00007766 000076B8 000083B2 * ..j.....c. *
00007708 000050  0000B4B0 00007782 00104001 00000008 * .....b. .... *
00007718 000060  C5E2D7E8 C3C2F3F9 F2F0F0F2 F0F7F0F1 * ESPYCB3920020701 *
00007728 000070  F1F4F4F5 F0F6F0F2 F0F1F0F2 00000000 * 144506020102.... *
00007738 000080  0000076C E4E82C4C 00A19C80 508C410F * ..%UY.<~.&... *
00007748 000090  810040C0 0C000000 80C10000 000000DD * a. ....A..... *
00007758 0000A0  000000B6 00000000 40404040 0008C5E2 * ..... ..ES *
00007768 0000B0  D7E8C3C2 F3F90500 00010000 B4000000 * PYCB39..... *
00007778 0000C0  0000FFFF FFB20000 76B80000 76F00000 * .....0.. *
00007788 0000D0  00080000 00060000 76F00000 B4000000 * .....0..... *
00007798 0000E0  B3D80000 00050000 00000000 00000000 * .Q..... *

```


When you press Enter, the DisAssembler screen displays. Scrolling forward is allowed until the end of the program; scrolling back is allowed to the point of start of disassembly.

```

----- StarTool DA : DisAssembler -----
COMMAND ==>>                               SCROLL ==>> PAGE

Selected Job= USER298A
Start Address= 000076B8 Area= PVT   pgm= ESPYCB39

Address  Offs  Machine Code  Opcde  Operand
000076B8 0000 47F0 F028      B      X'28' (,R15)
000076BC 0004                DC      X'00C3C5C5000001D8' .CEE...Q
000076C4 000C                DC      X'00000014'      ....
000076C8 0010 47F0 F001      B      1(,R15)
000076CC 0014                DC      X'98CEAC000000776E' Q.....>
000076D4 001C                DC      X'0000000000000000' .....
000076DC 0024                DC      X'00000000'      ....
000076E0 0028 90EC D00C      STM     R14,R12,12(R13)
000076E4 002C 4110 F038      LA      R1,X'38'(,R15)
000076E8 0030 98EF F04C      LM      R14,R15,X'4C'(R15)
000076EC 0034 07FF                BR      R15
000076EE 0036                DC      X'0000000076B80000' .....
000076FE 003E                DC      X'0000000091580000' ....J...
00007700 0040                DC      X'7766000076B80000' .....
00007706 004E                DC      X'83B20000B4B00000' C.....

```

SHOWCOBL

Displays the COBOL procedure division statement for a machine instruction. To execute this command, type SHOWCOBL on the command line; then, position the cursor on a machine instruction and press Enter. You can assign SHOWCOBL to a PF key.

In the following Storage Display panel, the SHOWCOBL command is typed on the command line. Then, the cursor is positioned on a machine instruction.

```

----- StarTool DA : Storage Display -----
COMMAND ==>> SHOWCOBL                               SCROLL ==>> PAGE

Selected Job= USER298A
Start Address= 0000770A Area= PVT   Program= ESPYCB39

Address  Offset      Data (Hexadecimal)      Data (EBCDIC)
0000770A 000D3A FA228138 813BF822 81388138 5830D0D4 * .a.a.8.a.a...M *
0000771A 000D4A 07F35820 905C58F0 202C4110 A5C805EF * .3...*.0...vH.. *
0000772A 000D5A D207D198 8148DC07 D198A2E0 D203D1A0 * K.Jqa...Jqs\K.J. *
0000773A 000D6A A47A4130 D1985030 D1A44110 D1A058F0 * u:.Jq&.Ju..J..0 *
0000774A 000D7A 210005EF 58309124 40F03000 5840D0D8 * .....j.0...Q *
0000775A 000D8A 07F45820 905C58F0 202C4110 A5B205EF * .4...*.0...v... *
0000776A 000D9A 4130813E 5030D198 41308138 5030D19C * .a.&.Jq..a.&.J. *
0000777A 000DAA 9680D19C D203D1A0 A4764130 A28C4140 * o.J.K.J.u...s.. *
0000778A 000DBA 91544150 D1985030 D1A45040 D1A85050 * j..&Jq&.Ju&Jy&& *
0000779A 000DCA D1AC4110 D1A058F0 210005EF 58309124 * J...J..0.....j. *
000077AA 000DDA 40F03000 5840D0DC 07F4D203 80206002 * 0...4K.... *
000077BA 000DEA D5038020 A2CD4780 B3E6D503 8020A174 * N...s...WN...~. *
000077CA 000DFA 4780B3E6 D5038020 A1554780 B3E6D503 * ...WN...~...WN. *
000077DA 000E0A 8020A2C9 4780B3E6 58F0202C 4110A59C * .sI...W.0...k.v. *
000077EA 000E1A 05EF58F0 202C4110 A58605EF 92F28150 * ...0...vf..k2a& *
000077FA 000E2A 5830D0B4 07F3D203 9170A0A0 58209170 * .....3K.j....j. *
0000780A 000E3A 1A28D201 81582720 D2039170 A09CD203 * ..K.a...K.j...K. *

```

When you press Enter, the COBOL Listing screen displays. The COBOL procedure division statement that corresponds to the machine instruction displays on the first line.

```

----- StarTool DA : COBOL Listing -----
COMMAND ===>                                SCROLL ===> PAGE

Selected Job= USER298A
Program=     ESPYCB39

ER STMT> 000436      add S0C7-BadField to S0C7-GoodField
000437      .
000438*-----*
000439 P03-PerformS806 section.
000440*-----*
000441      display MyProgramId ',P03,031I,S806 abend requested.'
000442      call S806-PgmName
000443      .
000444*-----*
000445 P04-PerformMult section.
000446*-----*
000447      display MyProgramId ',P04,041I,Multi-csect abend requested.'
000448      call 'espycb3a' using BadP04Data, S0C7-GoodField
000449      .
000450*-----*
000451 P10-ProcessJclParms section.
000452*-----*

```

Option 7 – Program Listing (COBOL only)

The Program Listing section is available only for COBOL. This panel displays the program source code and identifies the current statement. The current statement in the program in error is either the actual statement in error or the last call. The current statement in any program other than the program in error indicates the last known call in that program. The program source is merged into this panel from the source listing file (PDS) at view time or by the Language Server at abend time.

Use the FIND command on this panel. The syntax is:

```
F(ind) {variable} blank|previous d|p|blank
```

where

- *variable* is a Data Division data name, Procedure Division partial statement or a sequence number.
- *blank* searches forward; *previous* or *prev* searches in reverse direction.
- *d* means Data Division; *p* means Procedure Division; a *blank* means either Data Division or Procedure Division (the first occurrence of the variable displays).

In the following panel, the FIND command, `f badfield p`, searches for 'badfield' in the Procedure Division.

```

----- StarTool DA : COBOL Listing -----
COMMAND ===> f badfield p                               SCROLL ===> PAGE

Selected Job= USER298A
Program=      ESPYCB39

000026 environment division.
000027 input-output section.
000028 file-control.
000029     select Fd-Qsam-01 assign to Qsam0001
000030         file status is Qsam-01-FileStatus.
000031     select Fd-Qsam-02 assign to Qsam0002
000032         file status is Qsam-02-FileStatus.
000033     select Fd-Vsam0001 assign to Vsam0001
000034         organization is indexed
000035         access      is sequential
000036         record key  is Fd-Vsam0001-Key
000037         file status is Vsam0001-Fs1
000038         Vsam0001-Fs2.
000039     select Fd-Vsam0002 assign to Vsam0002
000040         organization is indexed
000041         access      is random
000042         record key  is Fd-Vsam0002-Key

```

The COBOL Listing panel is now positioned at the error statement 'add S0C7-BadField to S0C7-GoodField'. To view the contents of the operands, place the cursor on the line containing the error statement and press Enter.

```

----- StarTool DA : COBOL Listing -----
COMMAND ===>                                           SCROLL ===> PAGE

Selected Job= USER298A
Program=      ESPYCB39

ER STMT> 000436     add S0C7-BadField to S0C7-GoodField
000437     .
000438*-----*
000439 P03-PerformS806 section.
000440*-----*
000441     display MyProgramId ',P03,031I,S806 abend requested.'
000442     call S806-PgmName
000443     .
000444*-----*
000445 P04-PerformMult section.
000446*-----*
000447     display MyProgramId ',P04,041I,Multi-csect abend requested.'
000448     call 'espycb3a' using BadP04Data, S0C7-GoodField
000449     .
000450*-----*
000451 P10-ProcessJclParms section.
000452*-----*

```

A panel displays showing the contents of the fields (there is non-numeric data in S0C7-BadField).

```

----- StarTool DA : COBOL Listing -----
COMMAND ===>                                SCROLL ===> PAGE

Selected job= USER298A
Program=     ESPYCB39

----- StarTool DA : Procedure Division Variable -----
COMMAND ===>

Procedure Division statement:

000436      add S0C7-BadField to S0C7-GoodField

Statement variables and values:

Variable name : S0C7-BadField
Variable value:      828184

Variable name : S0C7-GoodField
Variable value:      00000C

Variable name :
Variable value:

```

If you enter the FIND command, `f badfield d`, the field displays within Working Storage. The storage content of each data item displays to the left of the definition. You can point-and-shoot on any Data Division item.

```

----- StarTool DA : COBOL Listing -----
COMMAND ===>                                SCROLL ===> PAGE

Selected Job= USER298A
Program=     ESPYCB39

000144 C:bad                05 S0C7-BadFieldC          pic x(3)
000145 P:828184            05 S0C7-BadField redefines S0C7-BadFieldC pic s
000146                    05 BadP04DataC.
000147 C:P04              10 BadP04DataC-Part1      pic x(3) value 'P04
000148 C:.                10 BadP04DataC-Part2      pic x(1) value x'0F
000149 P:D7F0F40F         05 BadP04Data redefines BadP04DataC      pic s
000150                    01 S806-AbendDataAreas.
000151 C:ESPYS806         05 S806-PgmName           pic x(8) value '
000152                    01 P05-DataAreas.
000153 C: .                05 P05-MainLoopCounter    pic s9(8) binary
000154                    01 ProgramSwitches.
000155 C:1                05 MainProgramStatus      pic x.
000156                    88 ProcessOk              value '1'.
000157                    88 ProcessDone            value '2'.
000158                    01 WS-MiscWorkAreas.
000159 C:AA              05 WS-WorkArea01 pic x(2).
000160                    *-----

```

A panel displays showing only the contents of that one data item at the time of theabend.

```

----- StarTool DA : COBOL Listing -----
COMMAND ===>                                SCROLL ===> PAGE

Selected Job= USER298A
Program=     ESPYCB39

----- StarTool DA : Data Division Variable -----
COMMAND ===>

Data Division Statement:

000144      05 S0C7-BadFieldC                pic x(3)      value 'bad'.

Variable Value at time of abend:

          bad

```

You can also point-and-shoot on Perform or Go To statements and DVS uses the LOCATE command to display the Paragraph or Section name referenced.

Option 8 – Data Management Control Blocks and Buffers

The Data Management Control Blocks and Buffers section provides information for every file open at the time of theabend. This section identifies:

- DDNAME
- Data set name
- Device type, volume serial number, disposition and access method
- Data Management Control Block information such as DCB, ACB, IOB, UCB, and so on
- Other information that is based on the type of file access method used
- Current and previous record information, when available. The last record read/written for every open file is viewable. All access methods are supported (BSAM, QSAM, VSAM, BPAM, and so on).

The following panel is the first to display when you choose Option 8. Type an S next to the file for which you want to view detail information.

```

----- StarTool DA : Data Management Control Blocks -----
COMMAND ==>>> SCROLL ==>> PAGE

Title: ABEND S0C7 IN JOB USER298A AT 15:03:07 ON SATURDAY, March 15, 2007

Jobname: USER298A Stepname: S010 Program Name: ESPYCB39
Abend: S0C7- Offset: 000D3A

Active Data Sets for this Job: (Select DDNAME for detail)

S DDNAME DEVICE UNIT VOLSER DSP ACCM EXCP Control Block Info
_ VSAM0002 3390 352 SMS011 SHR VSAM 0
_ VSAM0001 3390 352 SMS011 SHR VSAM 354
_ QSAM0002 SPOOL NEW QSAM 0 DSORG=PS,RECFM=FB
LRECL=00080,BLKSIZE=00000
_ QSAM0001 3390 352 SMS011 SHR QSAM 2 DSORG=PS,RECFM=FB
LRECL=00080,BLKSIZE=23440
_ SYSOUT SPOOL NEW QSAM 0 DSORG=PS,RECFM=FBA
LRECL=00121,BLKSIZE=12100
***** Bottom of data *****

```

If you select a QSAM file from the first panel, the current and previous record information displays (when available). The last record read/written is only available for open files.

```

----- StarTool DA : Data Management Control Blocks -----
COMMAND ==>>>

DDname: QSAM0001 Volser: SMS011 Unit Adr: Dev Type: 3390
Mode: GET LOCATE Excps: 2 Disp: SHR Acc Meth: QSAM
DSN: WST001.DA.USER.JCLLIB
DCB=( DSORG=PS,RECFM=FB,LRECL=00080,BLKSIZE=23440 )
SMSDATA: Storage Class= COMMON
Managmt Class= STANDEF
Data Class =
Current Record: (First 80 characters)
Address Offset D A T A E B C D I C
0005B1C0 0000 61615C60 60606060 60606060 60606060 //*------
0005B1D0 0010 60606060 60606060 60606060 60606060 -----
0005B1E0 0020 60606060 60606060 60606060 60606060 -----
0005B1F0 0030 60606060 60606060 60606060 60606060 -----
0005B200 0040 605C40E5 F5F1F040 F0F0F0F1 F0F0F0F0
Previous Record: (First 80 Characters)
Address Offset D A T A E B C D I C
** NO PREVIOUS RECORD AVAILABLE **

```

If you select a VSAM file from the first panel, VSAM ACB and RPL information is interpreted and displayed.

```

----- StarTool DA : Data Management Control Blocks -----
COMMAND ==>>

DDname: VSAM0001      Volser: SMS011      Access Method=VSAM  Excps: 354
DSN: WST001.DA.ES01.E390.HELP
File Summary:
Access type..... BASE CLUSTER
Dataset type..... KSDS
Processing type.. KEY,NFX,DDN,NDF,SEQ,NCI,IN,LEW,NIS,NRM,NRS,NSR,NUB

Last Request:
ID..... GET
Request ECB..... COMPLETED
Mode..... MOVE
Access..... SEQUENTIAL
Request Option1 (RPLOPT1).... SEQUENTIAL ACCESS IS BEING USED
Request Option2 (RPLOPT2).... LOCATE A RECORD BASED ON THE KEY
Current RBA..... 0080FE46

File errors: NONE

-----
| ESPM001S - Enter 'RECS' to display current and previous record data. |
-----

```

If you want to see the current and previous VSAM record information (when available), type RECS on the command line. The following panel displays.

```

----- StarTool DA : Data Management Control Blocks -----
COMMAND ==>> RECS

DDname: VSAM0001      Volser: SMS011      Access Method=VSAM  Excps: 354
DSN: WST001.DA.ES01.E390.HELP
File Summary:

----- StarTool DA : VSAM Records -----
COMMAND ==>>

Current Record: (First 80 characters)
Address  Offset          D A T A                      E B C D I C
00008878  0000    03B6E200 01000000 00000000 00010000  .¶S . .
00008888  0010    000040E2 E8E2E3C5 D440C1C2 C5D5C45E      SYSTEM ABEND;
00008898  0020    40E2F0F0 F1404040 40404040 40404040      S001
000088A8  0030    40404040 40404040 40404040 40404040
000088B8  0040    40404040 40404040 40404040 40404040
Previous Record: (First 80 Characters)
Address  Offset          D A T A                      E B C D I C
** NO PREVIOUS RECORD AVAILABLE **

```

Option C - ChangeMan ZMF Information

Option C is available only for programs under ChangeMan ZMF control.

The ChangeMan ZMF Information panel provides information for the load component and the package containing the abending program.

```

----- StarTool DA : ChangeMan ZMF Information -----
COMMAND ==>

General Package Information:
- Name           : TEST000021
- TITLE          : SAMPLE PACKAGE TO TEST DA
- LEVEL          : SIMPLE
- TYPE           : PLANNED PERMANENT
- STATUS         : DEVELOPMENT STATUS
- INSTALL Date   :

Load Component Information:
- Source Member  : IMSCOB
- Source TYPE    : SRC
- LOAD TYPE      : LOD

H      Display HISTORY Information
S      Display Source Listing
    
```

Select Option H to display history information for the component. All of the packages containing the component display.

```

----- StarTool DA : ChangeMan ZMF History Information -----
COMMAND ==>

HISTORY Information:
PACKAGE NAME  USERID  DATE      TIME      VERSION  MODIFICATION
*****
TEST000020   USER99  20070310  153423    01       05
TEST000021   USER99  20070312  100433    01       01
    
```


Select Option S to display the ChangeMan ZMF listing. The following example shows the program source.

```

Menu Utilities Compilers Help
-----
BROWSE      SYS02199.T153222.RA000.QAUSR07.ESPYT.H02   Line 00000105 Col 024 103
Command ==>                                         Scroll ==> PAGE
IDENTIFICATION DIVISION.                          00010000
PROGRAM-ID.   IMSCOB.                              00020000
                                                    00030000
*****
*                                                    00040000
*                                                    00050000
* SERENA STARTOOL DA, IMS/DA IVP, DL/I CBLTDLI      * 00060000
*                                                    00070000
*****
                                                    00080000
ENVIRONMENT DIVISION.                             00090000
CONFIGURATION SECTION.                            00100000
SOURCE-COMPUTER. IBM-390.                          00110000
OBJECT-COMPUTER. IBM-390.                          00120000
*                                                    00130000
*                                                    00140000
DATA DIVISION.                                    00150000
WORKING-STORAGE SECTION.                           00160000
                                                    00170000
01  PARM1 PIC S9(3) COMP-3 VALUE 1.                 00320000  BLW=
01  PARM2 PIC X(2) VALUE LOW-VALUE.                 00330000  BLW=
01  PARM2-R REDEFINES PARM2 PIC S9(3) COMP-3.      00340000  BLW=

```



NOTE For COBOL programs, the source can also be viewed in [Option 7 – Program Listing \(COBOL only\)](#).

Option D - Program Call/Trace Details

Option D gives the program call and trace details for the abending program.

The Call/Trace Detail panel gives information on the most recent calls made from the abending program.

```

----- ChangeMan ZMF-DA ----- COBOL Call/Trace Detail Row 1 to 2 of 2
Command ==>                                         SCROLL ==> PAGE

DVS Dump DSN: BHITFEI.ESPY.D110510.T042129.S001
Title: Abend U0929   Job:BGH#IMS2 at 04:21:29 on Tuesday   , May       10, 2011

***** Program Call/Trace Details *****
Csect/lmod *SYSTEM*/*SYSTEM* calls csect/lmod DFSDLB00/DFSDLBL0 at +X"   ".
Csect/lmod COMMON /DFSDLBL0 calls csect/lmod DFSLBLM0/DFSLBLM0 at +X"0FA2".
***** Bottom of data *****

```

Option L - Load a source listing to DA

Option L enables you to import a ChangeMan ZMF listing permanently into DA.

There are two main reasons you would want to do this:

- To pre-load a listing to the DBKF for later use in CICS.

- To load a listing from a non-ZMF data set.



NOTE The difference between option C and option L is as follows:

- Option C is a request to view a temporary copy of a listing downloaded from ZMF.
- Option L is a request to download a listing from ZMF and store a permanent copy of the listing in the DA repository.

When DA Batch retrieves a listing, it searches ZMF first. If it cannot find it, or if there is an error, it searches its own repository. This helps ensure that the correct listing is used.

The Source Listing Retrieval Menu gives several ways to load a listing into the DA Language Data Base.

- [Loading the Listing from a ZMF Server](#)
- [Loading the Listing from a Different Package](#)
- [Loading the Listing from a Different ZMF Listing Library](#)
- [Loading a Listing from a Different ZMF DSN](#)
- [Loading a Listing from a Non-ZMF Listing Source](#)

```

----- ChangeMan ZMF-DA ----- COBOL Source Listing Retrieval Menu -----
Command/Option ====>                                     Use PF3/END to exit
                                                           or PF1/HELP for help

Identify the listing to be retrieved and loaded into DA :
  Target module name ====>                               (Omit to use abending pgm from dump )

Identify the listing source :
  ZMF package name ====>                                 (Omit to use abending pgm's package )
  Workload Server SSN ====>                               (Omit to use default WorkLoad Server)
  Alternate listing DSN ====>                             .
  Is alternate listing DSN a ZMF library ? ====>         ( Y or N )
  ZMF subsystem ID ====>                                 (Overrides subsystem ID used at abend time).

Identify the target Language Data Base :
  CICS release number ====>                               520 CICS 5.2.0   610 TS 2.1   640 TS 3.1
  - DA-Batch - Leave blank.                               530 CICS 5.3.0   620 TS 2.2   650 TS 3.2
  - DA-CICS - Enter a valid release.                       630 TS 2.3   660 TS 4.1
                                                           670 TS 4.2

Jobcard image : (Note that the input is in both upper and lower case)
  Line 1 ====>                                           .
  Line 2 ====>                                           .
  Line 3 ====>                                           .

```

Loading the Listing from a ZMF Server

To load the ZMF listing for the abending program:

- Leave the **ZMF package name** field blank.
DA will extract the listing from the ZMF server from which the ZMF data at the time of the abend was extracted.
- You may request communication with an alternate ZMF server by specifying its subsystem identifier in the **ZMF subsystem ID** field.

Loading the Listing from a Different Package

To load the ZMF listing from a different package:

- Specify the requested package name in the **ZMF package name** field.
DA will extract the listing from the ZMF server from which the ZMF data at the time of the abend was extracted.
- You may request communication with an alternate ZMF server by specifying its subsystem identifier in the **ZMF subsystem ID** field.

Loading the Listing from a Different ZMF Listing Library

To load the ZMF listing from an alternate ZMF listing library:

- In the **Alternate listing DSN** field, specify the data set name of the alternate ZMF listing library from which DA can extract the listing.
- In the **Is alternate listing DSN a ZMF library?** field specify Y to indicate that the alternate listing data set is a ZMF library.
DA will extract the listing from the ZMF server from which the ZMF data at the time of the abend was extracted.
- You may request communication with an alternate ZMF server by specifying its subsystem identifier in the **ZMF subsystem ID** field.

Loading a Non-ZMF Listing

To load a non-ZMF listing:

- In the **Alternate listing DSN** field, specify the data set name of the non-ZMF listing data set. If the specified data set is a library, you must enter a member name.
- In the **Is alternate listing DSN a ZMF library?** field specify N to indicate that the alternate listing data set is not a ZMF library.
- Leave the **ZMF package name** and **ZMF subsystem ID** fields blank.
- In the **Target module name** field, specify the name of the program that is to be loaded into the DA Language Data Base.
If this field is blank, DA will use the abending program's name.
- In the **ZMF package name** field, specify the name of the ZMF package from which the listing is to be loaded.
If the **ZMF package name** field is blank and **Alternate listing DSN** is not specified, DA will treat the request as a request to load a ZMF listing and will attempt to extract the package name from the ZMF data in the dump.
- In the **Workload Server SSN** field, specify the name of the DA Workload Server through which you want to communicate with ZMF.
If the **Workload Server SSN** field is blank, the same Workload Server currently in use by your DA-Batch DVS session will be used.

Loading the Listing from a Different ZMF Package

Specifying a DSN in the **Alternate listing DSN** field overrides any listing.

To load the ZMF listing from a different package:

- In the **Alternate listing DSN** field, specify the DSN associated with the package name in the dump's ZMF data.

Loading a Listing from a Different ZMF DSN

To override a ZMF listing DSN:

- Specify the **Alternate listing DSN** and specify Y for the **Is alternate DSN a ZMF library?** field.

Note that no member name is required for a ZMF listing data set.

Loading a Listing from a Non-ZMF Listing Source

To specify a non-ZMF listing source:

- Specify the **Alternate listing DSN** and specify N for the **Is alternate DSN a ZMF library?** field. If the specified data set is a library, you must enter a member name.

The **Alternate listing DSN** field cannot be used in conjunction with the **ZMF package name** field. Use of the ZMF package name field implies that the listing DSN associated with a specific package is to be used.

- If the **Alternate listing DSN** is not enclosed in quotes, the entered DSN will be prefixed with the default DSN prefix value extracted from the ISPF variable ZPREFIX.
- To direct the listing to the DA-CICS target Language Data Base (LDB), in the **CICS release number** field specify one of the listed CICS release numbers.
- To direct the listing to the DA-Batch LDB, leave the **CICS release number** field blank.
- In the **Jobcard image** fields, specify up to 3 lines of text to be used as your jobcard for all submitted jobs. Note that the input is in both upper and lower case.

Multilevel Storage Viewing

The Debugging and Viewing Server (DVS) supports multilevel viewing for the point-and-shoot facility of the **Storage Display** panel. Your current position in the stack of multilevel displays for the **Display Storage** panel is indicated by a new **Display** counter in the upper right corner of the panel.

For example, let's say you use DVS to view a storage region beginning at address X'007EABF8'. The initial display in the multilevel display stack for the **Storage Display** panel is labeled **Display: 1 of 1** (shown in the example in red).

```

----- StarTool DA : Storage Display -----
COMMAND ==>                                     SCROLL ==> CSR

Selected Job= BGH9DM2
Start Address= 007EABF8 Area= PVT Program= TCB Display: 1 of 1

Address  Offset      Data (Hexadecimal)      Data (EBCDIC)
007EABF8 000000 007FC738 00006E18 007CB174 007C2FE8 * ."G...>..@...@.Y *
007EAC08 000010 00000000 00000000 7F640D90 80000000 * ..... "..... *
007EAC18 000020 0001FFFF 007CDCC0 007EA188 007CDC60 * .....@...=~h.@.- *
007EAC28 000030 00000001 007BDDDC 00000001 00FCC20C * .....#.....B. *
007EAC38 000040 018BFE00 81352054 007FC738 7F5F6DB4 * .....a...."G."~_ *
007EAC48 000050 00000044 007CECB4 FF5F6DB0 007EA188 * .....@...~_..=~h *
007EAC58 000060 81351BB2 007BDBE0 813526AC 807FC738 * a....#\a...."G. *
007EAC68 000070 00006F58 00000000 00000000 007EABF8 * ..?.....=.8 *
007EAC78 000080 00000000 007EA920 00000000 00000000 * .....=z..... *
007EAC88 000090 007EAE64 20000000 7FF16E54 7F644E08 * .=....."1>."+. *
007EAC98 0000A0 007FC7C8 807EA210 00000000 00002000 * ."GH.=s..... *
007EACA8 0000B0 00000000 007EACB4 00FCA038 00000000 * .....=..... *
007EACB8 0000C0 00000000 00000000 00000000 00000000 * ..... *
007EACC8 0000D0 007EAD50 00000000 7FFF9908 007EA920 * .=.&....".r...=z. *
007EACD8 0000E0 00000000 00000000 00000000 00000000 * ..... *
007EACE8 0000F0 00000040 7FFF9024 7F6448E0 00000000 * ... ". " . \ . . . *

```

From the first-level **Storage Display** panel above, you use the point-and-shoot facility to select address X'007FC738' at offset zero. The first-level display is pushed down on the display stack and the following newly selected storage area appears in the panel. It is marked in the **Storage Display** counter as **Display: 2 of 2**.

```

----- StarTool DA : Storage Display -----
COMMAND ==>                                     SCROLL ==> CSR

Selected Job= BGH9DM2
Start Address= 007FC738 Area= PVT Program= Display: 2 of 2

Address  Offset      Data (Hexadecimal)      Data (EBCDIC)
007FC738 000000 00000000 00000000 001EC022 00000000 * ..... *
007FC748 000010 075C0000 813526E0 00000000 007FC928 * *.a.\....."I. *
007FC758 000020 FF5F6DB0 007EA188 00000010 00000010 * ~_..=~h..... *
007FC768 000030 00000054 00000008 7F52C0A2 00000002 * ..... " . S . . . *
007FC778 000040 7F6429C8 0000000C 7F5F6EE8 03D5C442 * ".H...."->Y.ND. *
007FC788 000050 83D5B442 7F6428B0 00000008 FF5F6DA8 * cN..".....~_y *
007FC798 000060 0CE00F00 007BDBE0 00FECDD0 007CECB4 * \...#\.....@. *
007FC7A8 000070 00000001 007CECB4 FF5F6DB0 007EA188 * .....@...~_..=~h *
007FC7B8 000080 81351BB2 007BDBE0 0000001F 00000004 * a....#\..... *
007FC7C8 000090 007FC3B0 01352B4A 15000000 837FC738 * ."C....t....c"G. *
007FC7D8 0000A0 2400DB00 007FC7F8 974F109C 00000000 * ..... "G8p|..... *
007FC7E8 0000B0 00000000 C0010000 00000000 974F158C * .....p|.. *
007FC7F8 0000C0 00000000 00000000 00000000 007FC738 * ..... "G. *
007FC808 0000D0 E2E5D9C1 007CFAF0 7FFF9A00 00000000 * SVRA."0"..... *
007FC818 0000E0 00000000 00000000 00000000 00000000 * ..... *
007FC828 0000F0 00000000 00000000 00000000 00000000 * ..... *

```

You can continue to add as many displays as needed to the stack with successive point-and-shoot operations.

To navigate among the saved displays in the multilevel **Display Storage** stack:

- Press LEFT (PF10) to move backward (down) in the stack.
- Press RIGHT (PF11) to move forward (up) in the stack.

The stack contains 50 entries and is circular, with the last item linked to the first. If you continually press LEFT past the start of the stack, you will cycle to the last display and continue backward. If you continually press RIGHT past the end of the stack, you will cycle to the first display and continue forward. If you display more than 50 addresses, the 50th stack entry will be reused for each additional address.

Press END (PF3) to exit the **Display Storage** panel and delete the display stack.

Program Call/Trace Summary for Inactive Modules

The **Program Call/Trace Summary** panel **ESPYIPAA** (Option 5 on the **DVS Selection Menu**) displays all available information for any inactive programs found in the calling chain for the COBOL environment. These programs are marked "inactive" (IA) because they were called at some point during processing but have returned control. Drilldowns are possible on these entries, but the subsequent displays may be limited since storage and call information may have been released or cleared.

The following example shows two inactive modules on panel ESPYIPAA:

```

ESPYIPAA ----- StarTool DA : Program Call / Trace Summary   Row 1 to 8 of 8
COMMAND ===>                                               SCROLL ===> CSR

                Call Trace Summary and Application Program Attributes

                To display a called program's COBOL storage and data, enter "S" into the
                selection column and press enter. (Press PF1 for additional help).

---Called Pgm--- ---Called Pgm Attributes--- -----Caller----- ---Call----
S Program Load-Mod  Comp Date  Length  Language  Load-Mod Program  Offset Mode
- - - - -
_ ESPYCB39 ESPYCB39  04 AUG 2005  002B14  COBOL/ZOS  *SYSTEM*          CALL
_ ESPYCB39 ESPYCB39          CEEBINIT CEEBTOR  000000 CALL
_ IGZCFCC IGZCPAC   30 JUN 2003  000E00  HLA ASMBLR ESPYCB39 ESPYCB39  0017EA CALL
_ ESPYCB3B ESPYCB3B  04 AUG 2005  000674  COBOL/ZOS  IGZCPAC  IGZCFCC  0002CC CALL

----- INACTIVE -- MODULES -----
_ ESPYCB3D          04 AUG 2005          COBOL/ZOS          (IA)
_ ESPYCB3C          04 AUG 2005          COBOL/ZOS          (IA)
***** Bottom of data *****

```

Chapter 3

Mini Dump Debug Report

The Mini Dump Debug Report includes only the essential information you need to diagnose and solve an abend. The information is specific to the type of abend, the language of the failing program, and the types of files or databases in use. The report also shows the failing COBOL statement and the contents of the operands.

Mini Dump Sections	56
Mini Dump Report Example	57

Mini Dump Sections

The Mini Dump Debug Report contains the following sections:

- **Job and System Section:** Shows information about the environment of the abending program, including the operating system release level, date and time of the abend, job name, step name, and completion code. See [Job and System Section](#).
- **Diagnostic Section:** Includes a comprehensive explanation of the abend condition, specific information about the fields in the abending statement, and suggested courses of action (whenever possible). See [Diagnostic Section](#).
- **Failing Instruction Section:** Shows the location of the abending statement in the program, the instruction sequence prior to and including the abending instruction, the Program Status Word (PSW), the entry point of the module, the registers that apply to this abend, the location of the last I/O operation or subroutine call (if applicable), the compile date, and the length of the abending program. See [Failing Instruction Section](#).
- **Program Status Word and Register Section:** The Program Status Word and Register Section displays supporting environment data, such as Abend PSW, Program PSW, Load Module Name, Entry Point Address, Instruction Length Code (ILC), Load Point Address, Register contents (in both hex and decimal formats), and Register descriptions. See [Program Status Word and Registers Section](#).
- **Call Trace Section:** Provides a trace of all active programs that were executed using a call or link, and indicates the abending program. The program name, language, compile date, and length are also given for each application program. See [Call Trace Section](#).
- **COBOL Data Section:** Formats program storage for each application program in the calling chain. Included are the Task Global Table (TGT), Dynamic Storage Area (DSA), BLW cells, and BLL cells. See [COBOL Data Section](#).
- **Data Management Control Blocks:** Gives a status of each open file along with formatted current and previous records. See [Data Management Control Blocks Section](#).
- **ChangeMan ZMF Information:** Provides package and component information for programs under ChangeMan ZMF control. See [ChangeMan ZMF Information Section](#).
- **Specialized Database Support:** Provides database support for DB2 and IMS errors if these StarTool DA options are installed on the system.

Mini Dump Report Example

A COBOL S0C7 dump illustrates the different sections of the Mini Dump Report. This dump was generated by a COBOL/390 program running under LE.

Job and System Section

The Job and System Section provides standard information, regardless of the type of dump:

- Date of the error
- StarTool DA version number
- Completion code
- Job name
- Step name
- Job Owner
- Time of the error
- CPU model number
- Operating system release level
- Control Program FMID
- System on which your program was executing
- DFSMS release level
- JES2 release level

```

*****
*      Job Information      *
*****

Completion Code : S0C7-07
Jobname         : WST#9DM2
Stepname        : S010
Job Owner       : WSER21
Time            : 20:42:40

*****
*      System Information  *
*****

CPU MODEL       : 2066
Operating System : MVS/SP 7.0.2
Control Program FMID : HBB7705
SMF System ID   : D002
DFP              : V3.3.2
JES2            : z/OS 1.2

```

Diagnostic Section

The Diagnostic Section provides information to solve the problem. The cause of the error and corrective actions are described. The information varies, depending upon the programming language used and whether the error was external, data related, and so on.

```
*****  
*   Diagnostic Section   *  
*****
```

SYSTEM ABEND: S0C7

Description: This ABEND is a data exception and can only occur when decimal (packed) instructions are used. One of the following can cause this error:

- 1) The sign or digit codes of one or more bytes manipulated by the packed or CONVERT TO BINARY instructions is invalid for packed decimal use. Packed decimal digits must be in the range 0 through 9, with only the sign digit being a digit in the range A through F.
- 2) Fields in decimal (packed) arithmetic overlap incorrectly.
- 3) A packed decimal multiplicand has too many high-order significant digits.

Possible causes for this abend include:

```
Subscript error, referenced beyond table  
COBOL: working storage not initialized  
      bad data, should check data for errors  
      garbage in a field being tested or displayed  
      move zeroes to group level is display, had  
      sublevels that were not  
      period missing after imperative statements  
      within AT END clause  
      binary field in an arithmetic operation is not  
      large enough to accept result
```

User Action: Correct the format of the data being manipulated by the packed decimal instructions in the program and rerun the job.

Failing Instruction Section

The Failing Instruction Section provides the following information, depending upon the programming language used:

- Statement in error
- Failing Program name and offset
- Instruction sequence prior to and including abending instruction.
- Program's compile and link-edit dates
- Program-ID of compiler/assembler
- Load module name and load library name



NOTE When searching for the abending application load modules in private libraries (non-linklist and non-LPA) that are controlled by LLA (Library Lookaside), DA Batch always locates the most recent copy of the needed member.

- Location of the last I/O operation or subroutine call, if applicable
- Failing COBOL Statement Section - produced for COBOL programs when the source exists in either:
 - The Language Database
 - The Repository Data Set and the execution JCL contains the ESPYCOBL DD.



NOTE If you compile the program in ChangeMan ZMF, it must use the LIST and MAP options to see the failing COBOL statement.

```
*****
*           Failing Instruction Section           *
*****

The Failing Instruction is in Program (CSECT):
ESPYCB3A At Offset +X'00022E'

The failing instruction is:
===> AP      0(4,R7),0(4,R8)                <===

*****
*           Failing Instruction Analysis           *
*****
*
*           Instr Type = SS      Number of Operands = 2      *
BAD SIGN DIGIT--> * Operand 1 - Type = DATA  ADDR = 00009AD0 DATA = 00000C82      *
BAD DECIMAL DATA->* Operand 2 - Type = DATA  ADDR = 00009AD6 DATA = D7F0F40F      *
*****

*           Failing COBOL Statement Section       *
*****

**** The failing Instruction is within COBOL Statement:      **
===> 000034      add Field1 to Field2                        <==

**** Operand 1 ('TO' field) definition is as follows:
===> 01 Field2                                pic s9(7) comp-3.      <==
===> Contents:      00000C82

**** Operand 2 ('FROM' field) definition is as follows:
===> 01 Field1                                pic s9(7) comp-3.      <==
```

==> Contents: D7F0F40F

```
*****
*          Instruction sequence prior to and including abending instruction.          *
*                                                                                       *
* Address      Offs      Mach Code          Opcde      Operand          Data          *
* -----      -      -      -      -      -      -      -      -      -      -      *
* 0004C16A     001F2     1882                LR           R8,R2                *
* 0004C16C     001F4     5820 1004                L            R2,4(,R1)            *
* 0004C170     001F8     4120 2000                LA           R2,0(,R2)            *
* 0004C174     001FC     5020 9134                ST           R2,X'134'(,R9)      *
* 0004C178     00200     1872                LR           R7,R2                *
* 0004C17A     00202     D203 90EC A010          MVC          X'EC'(4,R9),X'10'(R10) *
* 0004C180     00208     BF2F 9148                ICM          R2,15,X'148'(R9)    *
* 0004C184     0020C     58B0 C008                L            R11,8(,R12)          *
* 0004C188     00210     4780 B000                BZ           0(,R11)              *
* 0004C18C     00214     5830 905C                L            R3,X'5C'(,R9)        *
* 0004C190     00218     58F0 30F4                L            R15,X'F4'(,R3)       *
* 0004C194     0021C     4110 A07E                LA           R1,X'7E'(,R10)       *
* 0004C198     00220     05EF                BALR         R14,R15              *
* 0004C19A     00222     5A20 C000                A            R2,0(,R12)           *
* 0004C19E     00226     5020 9148                ST           R2,X'148'(,R9)      *
* 0004C1A2     0022A     9640 9138                OI           X'138'(R9),X'40'    *
* 0004C1A6     0022E     FA33 7000 8000          AP           0(4,R7),0(4,R8)    *
*****
```

The program was compiled on 09 Sep 2002 By PGMID 5648-A25-00, V 2.1 M 02 (COBOL/MVS)
 It is a CSECT in Load Module ESPYCB3A Compile Time Was:
 The module was loaded from STEPLIB Library WST001.DA.E554.LNKLIB
 It was Link Edited on 09 Sep 2002

Program Status Word and Registers Section

The PSW and Registers Section displays Supporting Environmental Data, such as:

- Abending Program PSW (Program Status Word)
- Program PSW
- Load Module Name
- Entry Point Address
- Instruction Length Code (ILC)
- Load Point Address
- Register contents in both hex and decimal formats, and register descriptions

```

*****
* PSW and Register Section *
*****
ABEND PSW - 078D1000 8004C1AC  A(ESPYCB3A) + 000234
Prog PSW - 078D1000 8E77D428  A(ESPY9TRM) + 000428
Load Module - ESPYCB3A                      Entry Point Address - 0004BF78
ILC - 06,  INTC - 07                      Load Point Address - 0004BF78

Registers at time of error (Descriptions based on resolutions as 31 bit addresses).

Register      : 00
Hex Value     : 0E744498
Decimal Value : 242,500,760
Points to    : Address 0E740000 + 004498 (Private Area, Subpool 001, ALLO)

Register      : 01
Hex Value     : 0E7441F0
Decimal Value : 242,500,080
Points to    : Address 0E740000 + 0041F0 (Private Area, Subpool 001, ALLO)

Register      : 02
Hex Value     : 00000001
Decimal Value : 1
Points to    : Load Point of PSA + 000001

Register      : 03
Hex Value     : 0004BF78
Decimal Value : 311,160
Points to    : Load Point of ESPYCB3A + 000000

Register      : 04
Hex Value     : 0004BFB0
Decimal Value : 311,216
Points to    : Load Point of ESPYCB3A + 000038

Register      : 05
Hex Value     : 0001B1BC
Decimal Value : 111,036
Points to    : Address 0001B000 + 0001BC (Private Area, Subpool 001, ALLO)

Register      : 06
Hex Value     : 0E7416A0
Decimal Value : 242,488,992
Points to    : Address 0E740000 + 0016A0 (Private Area, Subpool 001, ALLO)

Register      : 07
Hex Value     : 00009AD0
Decimal Value : 39,632
Points to    : Load Point of ESPYCB39 + 002418

Register      : 08
Hex Value     : 00009AD6
Decimal Value : 39,638
Points to    : Load Point of ESPYCB39 + 00241E

Register      : 09
Hex Value     : 0004C218
Decimal Value : 311,832
Points to    : Load Point of ESPYCB3A + 0002A0

Register      : 10
Hex Value     : 0004C080
Decimal Value : 311,424
Points to    : Load Point of ESPYCB3A + 000108

Register      : 11
Hex Value     : 0004C19A
Decimal Value : 311,706

```

Points to : Load Point of ESPYCB3A + 000222
Register : 12
Hex Value : 0004C074
Decimal Value : 311,412
Points to : Load Point of ESPYCB3A + 0000FC

Register : 13
Hex Value : 0E7443F0
Decimal Value : 242,500,592
Points to : Address 0E740000 + 0043F0 (Private Area, Subpool 001, ALLO)

Register : 14
Hex Value : 0004C11C
Decimal Value : 311,580
Points to : Load Point of ESPYCB3A + 0001A4

Register : 15
Hex Value : 8004C128
Decimal Value : -2,147,172,056
Points to : Load Point of ESPYCB3A + 0001B0

Call Trace Section

The Call Trace Section displays the programs that were called and in what order. Included are the Program Attributes for all application programs in the Call Trace Summary.

```

*****
*   Call Trace Section   *
*****

**** Caller ****      ***** At *****      ***** Calls *****
Load-Mod Program      Offset      -- Calls Program Load-Mod
*SYSTEM*
CEEBINIT CEEBTOR      + 0000      -- Calls ESPYCB39 ESPYCB39
ESPYCB39 ESPYCB39    + 1038      -- Calls IGZCFCC  IGZCPAC
IGZCPAC  IGZCFCC     + 02DA      -- Calls ESPYCB3A ESPYCB3A
ESPYCB3A IGZCBSO     + 036E      -- Calls IGZEINI  IGZEINI

*****
*
*   Program Attributes   *
*
Load-Mod Program      Compile Date      Length      Language
ESPYCB39 ESPYCB39    09 Sep 2002    003D08     COBOL/MVS
ESPYCB39 ESPYCB39    09 Sep 2002    003D08     COBOL/MVS
IGZCPAC  IGZCFCC     06 Dec 2001    000E00     COBOL/MVS
ESPYCB3A ESPYCB3A    09 Sep 2002    000448     COBOL/MVS
IGZEINI  IGZEINI     06 Dec 2001    0019B0     COBOL/MVS

```

COBOL Data Section

The COBOL Data Section contains the formatted program storage for all COBOL programs in the calling chain. Included are the Task Global Table (TGT), Dynamic Storage Area (DSA), BLW cells, and BLL cells.

* COBOL Data Section *

COBOL data summary for program ESPYCB3A :

TGT address : 0004C218 Working-Storage address : 0004C3B8 Working-Storage length : 00000008
DSA address : 0E7443F0 Previous DSA address : 0E744208
DSA register save area :
R0 8004C6AC R1 0004C218 R2 0E7441F0 R3 000092A4 R4 0004BF00 R5 0001B1BC R6 0E7416A0 R7 00FC69C0
R8 0001BA80 R9 0E7444AC R10 0E740100 R11 0004C4B0 R12 0E50A828 R13 0E7443F0 R14 8004C81E R15 8001C038

COBOL data summary for program ESPYCB39 :

TGT address : 00009158 Working-Storage address : 00009990 Working-Storage length : 00001A30
DSA address : 0E744030 Previous DSA address : 0E50B1A0
DSA register save area :
R0 0E744208 R1 0E7441F8 R2 0001B7FC R3 00007B16 R4 000092A4 R5 0E7441F0 R6 0E508F70 R7 00026060
R8 00009990 R9 00009158 R10 000077EC R11 00008320 R12 000077B4 R13 0E744030 R14 800086F0 R15 8E6D8308

----- COBOL control blocks for program ESPYCB3A -----

--- DSA (LE Dynamic Storage Area) for program ESPYCB3A ---

0E7443F0 +0000 00104001 0E744208 0E7444E0 8004C81E 8001C038 8004C6AC 0004C218 0E7441F0 \..H.....F...B...0
0E744410 +0020 000092A4 0004BF00 0001B1BC 0E7416A0 00FC69C0 0001BA80 0E7444AC 0E740100 ..ku.....
0E744430 +0040 0004C4B0 0E50A828 00000000 0E744498 00000000 00000000 0E7443F0 0004C218 ..D..&y.....q.....0..B.
0E744450 +0060 0E50A828 00000000 00000000 00000000 00000000 00000000 00000000 00000000 &y.....

--- TGT (Task Global Table) for program ESPYCB3A ---

0004C218 +0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0004C238 +0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0004C258 +0040 00000000 00000000 00000000 F3E3C7E3 00000000 05000000 66020220 0E740100 0001B7FC3TGT.....
0004C278 +0060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0004C298 +0080 0E50A828 0000014C 00000000 00000000 00000000 00000001 E2E8E2D6 E4E34040 &y.....<.....SYSOUT
0004C2B8 +00A0 C9C7E9E2 D9E3C3C4 00000000 00000000 00000000 00000000 00000000 IGZSRTC.....
0004C2D8 +00C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0004C2F8 +00E0 00000000 00000000 0004C074 00000001 0004C350 0E7416A0 0004C09C 00000000C&.....
0004C318 +0100 0004BF78 0004C088 0004C350 0004C07C 00000000 0004C3B8 00000000 00000000h..C&...@...C.....
0004C338 +0120 00000000 0004C368 0004C3B8 00000000 00009AD6 00009AD0 40000000 00000000C...C.....0.....
0004C358 +0140 00000000 00000000 00000001

--- Working storage --- BLW cell # 1 for program ESPYCB3A --- Address X"0004C3B8" --- Length X"00000008" ---

0004C3B8 +0000 C5E2D7E8 C3C2F3C1 ESPYCB3A

--- Linkage section --- BLL cell # 1 for program ESPYCB3A --- Address X"00009AD6" --- Length X"00000040" ---

00009AD6 +0000 D7F0F40F 00000000 0000C5E2 D7E8E2F8 F0F60000 00050000 0000F100 00000000 P04.....ESPYS806.....1.....
00009AF6 +0020 0000C1C1 4EF5F24B F1F0C54E F2F74EF5 F2F2F0C5 60F2F700 000052E3 AEB50000 ..AA+52.10E+27+5220E-27...T.....

--- Linkage section --- BLL cell # 2 for program ESPYCB3A --- Address X"00009AD0" --- Length X"00000040" ---

00009AD0 +0000 00000C82 8184D7F0 F40F0000 00000000 C5E2D7E8 E2F8F0F6 00000005 00000000 ...badP04.....ESPYS806.....
00009AF0 +0020 F1000000 00000000 C1C14EF5 F24BF1F0 C54EF2F7 4EF5F2F2 F0C560F2 F7000000 1.....AA+52.10E+27+5220E-27...
S t a r T o o l D A PAGE 008

----- COBOL control blocks for program ESPYCB39 -----

--- DSA (LE Dynamic Storage Area) for program ESPYCB39 ---

0E744030 +0000 00104001 0E50B1A0 00048B28 800086F0 8E6D8308 0E744208 0E7441F8 0001B7FC&.....f0.._c.....8....
0E744050 +0020 00007B16 000092A4 0E7441F0 0E508F70 00026060 00009990 00009158 000077EC ..#...ku...0.&...--r...j.....
0E744070 +0040 00008320 000077B4 00000000 0E744208 00000000 00000000 0E744030 00009158 ..C.....j.....
0E744090 +0060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 ..C.....q.....

--- TGT (Task Global Table) for program ESPYCB39 ---

00009158 +0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00009178 +0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00009198 +0040 00000000 00000000 00000000 F3E3C7E3 00000000 05000000 66020220 0E740100 0001B7FC3TGT.....
000091B8 +0060 00000000 00000004 00001A30 00000000 00000000 00048108 00000000 00000000a.....
000091D8 +0080 0E50A828 000001B8 00000000 00000000 00000000 00000001 E2E8E2D6 E4E34040 &y.....<.....SYSOUT
000091F8 +00A0 C9C7E9E2 D9E3C3C4 00000000 00000000 00000000 00000000 00000000 IGZSRTC.....
00009218 +00C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00009238 +00E0 00000000 00000000 000077B4 00000001 000092FC 0E7403F0 00007D6A 00000000k...0...'
00009258 +0100 00007688 00007860 000092A8 000077E8 000092EC 00009990 00000000 00000000ky...Y..k...r.....
00009278 +0120 00000000 00009940 00009990 0000A990 00000000 0E508F70 00000000 000311C0r...r...z.....&.....
00009298 +0140 00026060 00009870 000098D8 0E7416A0 00000000 00000008 00000006 000000BB ..--q...qQ.....
000092B8 +0160 00000058 00000006 00000002 00000000 00000000 00000000 00000001 00000001q.....
000092D8 +0180 00000000 00000000 00000000 00000000 00000000 00009370 000094D0 000095D0l...m...n.
000092F8 +01A0 000096D0 40000000 00000000 00000000 000097D0 00000001o.....p.....

--- Working storage --- BLW cell # 1 for program ESPYCB39 --- Address X"00009990" --- Length X"00001000" ---


```

00009990 +0000 C5E2D7E8 C3C2F3F9 F0F0F0F0 F0F00000 00000000 F2F30008 00000010 00000000 ESPYCB39000000.....23.....
000099B0 +0020 00000000 00000000 D4E4D3E3 C9000000 E5A28194 F0F0F0F2 E5E2C1D4 40998581 .....MULTI...Vsam0002VSAM rea
000099D0 +0040 849585A7 A3F0F0F2 F3000800 000010E3 8889A240 89A240D8 E2C1D4F0 F0F0F240 dnext0023.....This is QSAM0002
000099F0 +0060 93899585 40F14B40 40404040 40404040 40404040 40404040 40404040 line 1.
00009A10 +0080 40404040 40404040 40404040 40404040 40404040 40404040 404040E3 T
00009A30 +00A0 8889A240 89A240D8 E2C1D4F0 F0F0F240 93899585 40F24B40 40404040 40404040 his is QSAM0002 line 2.
00009A50 +00C0 40404040 40404040 40404040 40404040 40404040 40404040 40404040
00009A70 +00E0 40404040 40404040 40404040 404040E3 8889A240 89A240D8 E2C1D4F0 F0F0F240 This is QSAM0002
00009A90 +0100 93899585 40F3404D A3888540 9381A2A3 40938995 855D4B40 40404040 40404040 line 3 (the last line).
00009AD0 +0120 40404040 40404040 40404040 40404000 40404040 40404040 40404000
00009AD0 +0140 00000C82 8184D7F0 F40F0000 00000000 C5E2D7E8 E2F8F0F6 00000005 00000000 ...badP04.....ESPYS806.....
00009AF0 +0160 F1000000 00000000 C1C14EF5 F24BF1F0 C54EF2F7 4EF5F2F2 F0C560F2 F7000000 1.....AA+52.10E+27+5220E-27...
00009B10 +0180 52E3AEB5 00000000 5392074A 78E69C7F 52E3AEB5 00000000 5392074A 78E69C7F .T.....k.c.W."..T.....k.c.W."
00009B30 +01A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00009B50 +01C0 000000F3 F4F5C6F2 F3F4D5F2 F3F4C5F3 F4F5C6F2 F3F4D5F2 F3F4C5C3 F4F5F6D2 ..345F234N234E345F234N234EC456K
00009B70 +01E0 F3F4F5C2 F3F4F5F4 F3F4F5F6 60F2F3F4 F54EF2F3 F4F5F3F4 F5F64EF2 F3F4F560 345B345+3456-2345+23453456+2345-
00009B90 +0200 F2F3F4F5 4EF2F3F4 F5000000 00F0F1F0 F2F3F3F0 F3000000 3D02FA03 15000024 2345+2345....01023303.....
00009BB0 +0220 90000206 99000206 9A000211 66000211 661D001D 001D0000 1D00788C 00050037 .....r.....
00009BD0 +0240 00FF00FF 000000FF 00000000 00000000 00000000 00000000 00000000
00009BF0 +0260 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00009C10 +0280 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00009C30 +02A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00360Th
00009C50 +02C0 89A24040 40404040 A2404040 40404040 40408140 40404040 40404040 40404040 is is a
00009C70 +02E0 40A385A2 A34B4040 40404040 40404040 00000000 00000000 000000C1 00000000 test. ....A....
00009C90 +0300 00000000 00000000 00C20000 00000000 000000C3 00001C00 0000C300 0000C300 .....B.....C.....C....C.
00009CB0 +0320 0000F1F2 F3F4F5F6 F7F8F9F0 F1F2F3F4 F5F6F7F8 F9F0F1F2 F3F4F5F6 F7F8F9F0 ..123456789012345678901234567890123456789012
00009CD0 +0340 F1F2F3F4 F5F6F7F8 F9F0F1F2 F3F4F5F6 F7F8F9F0 F1F2F3F4 F5F6F7F8 F9F0F1F2 12345678901234567890123456789012
00009CF0 +0360 F3F4F5F6 F7F8F9F0 F1F2F3F4 F5F6F7F8 F9F0F1F2 F3F4F5F6 F7F8F9F0 F1F2F3F4 34567890123456789012345678901234
00009D10 +0380 F5F6F7F8 F9000000 00000000 00000000 00000000 00000000 00000000 00000000 56789.....
00009D30 +03A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000
S t a r T o l D A PAGE 009

00009D50 +03C0 - +043F 00009DCF same as above
00009DD0 +0440 D1968895 40404040 40404040 404040C4 96854040 40404040 40404040 40404040 John Doe
00009DF0 +0460 40404040 40404040 E3889694 81A24040 40404040 40404000 00000000 00000000 Thomas
00009E10 +0480 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00009E30 +04A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00009E50 +04C0 00000000 00000000 00000000 000000C9 95A39996 84A483A3 9699A840 C3968296 .....Introductory Cobo
00009E70 +04E0 93404040 40404040 40404040 40F0F0F0 F1F1F100 00000000 00F9F600 00000000 l 000111.....96.....
00009E90 +0500 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00009EB0 +0520 - +071F 0000A0AF same as above
00009ED0 +0540 00000000 00000000 00000000 00000000 C1C1C2C2 C3C3C2F2 D8E6D8E6 C5D9E3E8 .....AABBCCB2QWQWERTY
00009EF0 +0560 00000000 F0F1F5C0 00000000 00000000 00000000 00000000 00000000 .....015.....
00009F10 +0580 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00009F30 +05A0 - +0FDF 0000A96F same as above
00009F50 +05C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00009F70 +05E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000

--- Working storage --- BLW cell # 2 for program ESPYCB39 --- Address X"0000A990" --- Length X"0000A30" ---

0000A990 +0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 .....
0000A9B0 +0020 - +0A1F 0000B3AF same as above
0000A9D0 +0040 00000000 00000000 00000000 00000000 .....

--- Linkage section --- BLL cell # 1 for program ESPYCB39 --- Address X"0E508F70" --- Length X"00000007" ---

0E508F70 +0000 0005D4E4 D3E3C9 .....MULTI
--- Linkage section --- BLL cell # 2 for program ESPYCB39 --- Address X"00000000" --- Length X"00000000" ---

00000000 +0000 <<< Display bypassed due to storage address being zero >>>

```

Data Management Control Blocks Section

The Data Management Control Blocks Section provides information for every file open at the time of the abend. This section identifies:

- DDNAME
- Data set name
- Device type, volume serial number, disposition and access method
- Data Management Control Block information such as DCB, ACB, IOB, UCB, and so on
- Other information that is based on the type of file access method used
- Current and previous record information (when available)

```

*****
* Data Management Control Blocks for DDNAME - VSAM0001 *
*****
                DSNAME: WST001.DA.ES54.E390.HELP
                Device Type: 3390
                VOL-SER: SRSM71
                Unit Address: 471
                DISP: SHR
                Access Method: VSAM

*** S M S D A T A : ***
Storage Class = COMMON
Management Class = STANDEF
Data Class =

                File Access Type: BASE CLUSTER
                Dataset Type: KSDS
                Processing Options: CNV,NFX,DDN,NDF,NCI,LEW,NIS,NRM,NRS,NSR,NUB
                ACB Address: 00048378
                RPL Address: 000483C8

*** ACB Return Codes: Return Code = 000 (X'00')
                    Request completed successfully.
                    Reason Code = 000 (X'00')

*** RPL Return Codes: Return Code = 000 (X'00')
                    Request completed successfully.
                    Reason Code = 000 (X'00')

*** Current Record ***
| ADDR | DISP |----- D A T A -----|----- TRANSLATION -----|
00009870 00000 03B6E200 01000000 00000000 00010000 000040E2 E8E2E3C5 D440C1C2 C5D5C45E | S          SYSTEM ABEND;
00009890 00020 40E2F0F0 F1404040 40404040 40404040 40404040 40404040 40404040 40404040 | S001
000098B0 00040 40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040
000098D0 00060 40000000

                Last Request: Read
                ID: GET
                Request ECB: Completed
                Mode: MOVE
                Access: Sequential
                Request Option1 (RPLOPT1): Sequential Access is being used
                Request Option2 (RPLOPT2): Locate a record based on the key
                Current RBA: 00000000

*****
* Data Management Control Blocks for DDNAME - QSAM0001 *
*****
                DSNAME: WST001.DA.ES54.JCLLIB
                Device Type: 3390
                VOL-SER: SRSM4F
                Unit Address: 44F
                DISP: SHR
                Access Method: QSAM
                File EXCP Count: 0000002
                MODE: GET LOCATE
                Block Count: N/A
                DCB: (DSORG=PS,RECFM=FB,LRECL=00080,BLKSIZE=23440)

*** S M S D A T A : ***
Storage Class = COMMON
Management Class = STANDEF
Data Class =

*** D E B ***
| ADDR | DISP |----- D A T A -----|----- TRANSLATION -----|
008CB0DC 00000 038F61A8 108CB298 68000000 00001100 01000000 FF000000 8F009310 048CB0B8 | /y q          l
008CB0FC 00020 50F350B0 0000036A 00000376 000E00C3 00010001 00000000 00000000 00000050 | &3&          C          &
008CB11C 00040 F3C2C1D8 C1C10000 00000000 00000000 00000010 00540002 008BEAC8 24019EE0 | 3BAQAA          H \
008CB13C 00060 00000000 40004800

*** D C B ***

```

```

| ADDR | DISP |----- D A T A -----|----- TRANSLATION -----|
00009310 00000 0E740448 00000000 036A0005 20F3103C 002FE5A2 0502B628 00004000 00006AE0 | 3 Vs |
00009330 00020 C6022626 900238C8 00404800 008CB0DC 12E33E48 00E2DD10 0A022806 02095B90 | F H T S |
00009350 00040 30013030 00006BA0 00031760 000311C0 00000050 00000000 00000000 00D85690 | , - & Q |
*** U C B ***
| ADDR | DISP |----- D A T A -----|----- TRANSLATION -----|
00F350B0 00000 01A8FF8C 044F0000 00000000 00F4F4C6 3030200F 00F35088 000F0100 E2D9E2D4 | y | 44F 3&h SRSM |
00F350D0 00020 F4C61004 00000009 38688077 00000000 00000000 00000000 D9000D0C 0D1A2424 | 4F | R |
00F350F0 00040 15BE0000 00130000 00FC113C 900170F4 50152400 00010000 00000000 00000000 | | 4& |
00F35110 00060 00000940 202A0008 020217E0 00000000 00000000 00FC0FF4 00F350D8 00000000 | | 4 3&Q |
*** I O B - Prefix ***
| ADDR | DISP |----- D A T A -----|----- TRANSLATION -----|
00006BA0 00000 20006BF0 7F000000 | | ,0" |
*** I O B - Standard Section ***
| ADDR | DISP |----- D A T A -----|----- TRANSLATION -----|
00006BA8 00000 02000000 00006BA4 00006BD8 0C4055F0 00000000 00009310 00000000 00000000 | | ,u ,Q 0 l |
00006BC8 00020 00000003 6A000520 060311C0 A0005B90 | | | $ |
*** I O B # 001 ***
| ADDR | DISP |----- D A T A -----|----- TRANSLATION -----|
00006BF0 00000 20006C40 41000000 02000000 00006BF4 00006C28 0D005B90 00000000 00009310 | % | ,4 % $ l |
00006C10 00020 00000000 00000000 00000003 6A000521 | | | |
*** I O B # 002 ***
| ADDR | DISP |----- D A T A -----|----- TRANSLATION -----|
00006C40 00000 20006C90 00000000 02000000 00006C44 00000000 00000000 00000000 00009310 | % | % l |
00006C60 00020 00000000 00000000 00000003 6A000522 | | | |
S t a r T o o l D A PAGE 014
*** I O B # 003 ***
| ADDR | DISP |----- D A T A -----|----- TRANSLATION -----|
*** I O B # 003 ***
| ADDR | DISP |----- D A T A -----|----- TRANSLATION -----|
00006C90 00000 20006B50 00000000 02000000 00006C94 00000000 00000000 00000000 00009310 | ,& | %m l |
00006CB0 00020 00000000 00000000 00000003 6A000520 | | | |
*** I O B # 004 ***
| ADDR | DISP |----- D A T A -----|----- TRANSLATION -----|
00006B50 00000 01006BA0 00000000 02000000 00006B54 00000000 00000000 00000000 00009310 | | | |
00006B70 00020 00000000 00000000 00000003 6A000520 | | | |
*** Current Record ***
| ADDR | DISP |----- D A T A -----|----- TRANSLATION -----|
000311C0 00000 61615C60 60606060 60606060 60606060 60606060 60606060 60606060 60606060 | /*----- |
000311E0 00020 60606060 60606060 60606060 60606060 60606060 60606060 | |----- |
00031200 00040 605C40E5 F5F3F040 F0F0F0F1 F0F0F0F1 | | - * V530 00010001 |
*** Last Record Read
| ADDR | DISP |----- D A T A -----|----- TRANSLATION -----|
00031710 00000 61615C85 96864040 40404040 40404040 40404040 40404040 | /*eof |
00031730 00020 40404040 40404040 40404040 40404040 40404040 40404040 | | | |
00031750 00040 404040E5 F5F3F040 F0F0F1F8 F0F0F0F1 | | V530 00180001 |

```

ChangeMan ZMF Information Section

The ChangeMan ZMF Information Section displays programs under ChangeMan ZMF control. It includes:

- Details for the package containing the abending program
- Component information
- History for all packages containing the component

```
*****
*   ChangeMan ZMF   *
*****
```

```

      PACKAGE ID: STDA000004      STATUS: DEV      INSTALL DATE: 20071215
COMPLEX/SUPER ID: N/A          STATUS:
PACKAGE TITLE
====> STDA : User Test Package
APPLICATION                     ====> STDA
REQUESTER'S NAME                 ====> User
REQUESTER'S PHONE                ====> 5555
WORK REQUEST ID                  ====>
DEPARTMENT                       ====> QA
PACKAGE LEVEL                     ====> 1          (1-Simple, 2-Complex,
                                           (3-Super, 4-Participating)
PACKAGE TYPE                       ====> PLANNED    (Planned or Unplanned)PACK
PACKAGE TIME SPAN                  ====> PERM      (Permanent or Temporary)
UNPLANNED REASON CODE             ====>
                                           S t a r T o o l   D A
TEMPORARY CHANGE DURATION =====>
```

```

      PACKAGE ID: STDA000004      STATUS: DEV      INSTALL DATE: 20071215
SOURCE                           LOAD
NAME   TYPE   NAME   TYPE STATUS   PROMOTION   CHANGED   ID
ESPYCOB3 SRC  ESPYCOB3 LOD  ACTIVE      20070618 091416 USER18

PACKAGE ID STA  PROMOTION VV.MM  LAST ACTION   SIZE  PROCNAME  ID
STDA000004 DEV          01.03 2007/06/18 09:14 00095 CMNCOB2  USER18
```

Chapter 4

COBOL Source-Level Support

StarTool DA provides source-level support for debugging COBOL programs. This support is available with both the Mini Dump Debug Report and the Debug/Viewing Server (DVS). StarTool DA allows flexibility by providing production mode and development processing.

The following topics contain more information on StarTool DA source-level support:

Processing Modes	70
Language Environment	72
Debugging and Viewing Server Call/Trace Summary for COBOL Programs	73

Processing Modes

StarTool DA provides both [Production Mode](#) and [Development Mode](#) processing for source-level support when debugging COBOL programs.



NOTE For both modes, you must compile your program with the SOURCE, LIST, MAP, XREF, and NOOFFSET options. For VS-COBOL, use PMAP instead of LIST, and DMAP instead of MAP.

Production Mode

In Production Mode, compile your program using the StarTool DA Language Server. This requires adding two StarTool DA language processor steps to your compile JCL. Sample JCL for these steps is provided on the StarTool DA product tape. Production Mode features include:

- A method for storing source output
- After-the-abend compilation of abended programs
- Source-level analysis
- Access to Working Storage areas using COBOL definitions
- Support for different versions of COBOL

A method for storing source output

The StarTool DA Language Server writes program source to the Language Database. There is an individual VSAM file for each program.

The Language Database is accessed automatically during abend time to produce the Failing Instruction Section of the Mini Dump. It is also accessed by the Debug/Viewing Server during viewing.

After-the-abend compilation of abended programs

The Language Server allows you to generate after-the-abend compiles and use them with the Debug/Viewing Server (DVS). This means that you can smoothly migrate over time to a complete production language database.

Source-level analysis

You can see the source that corresponds to a set of instructions. For example, for a COBOL program and a given hex area in a dump, you can look at the assembler, the COBOL source from the compiler, and the COBOL input to the translator. This source can be an SQL source statement or a simple COBOL verb.

Access to Working Storage areas using COBOL definitions

You can look at the contents of any Working Storage area in the dump using the COBOL definition. The values of Working Storage are frozen at abend time. Uninitialized or invalid data areas, indexes, and subscripts are easily found.

Support for different versions of COBOL

COBOL II, LE/370 (SAA AD/CYCLE COBOL/370), LE/390, and COBOL for MVS are supported.

Development Mode

Using Development Mode in a program development environment is attractive because you need not run the full language database compilation to make the compiler output available for StarTool DA Batch. You use the COBOL compiler output directly by writing the SYSPRINT output to either a sequential or partitioned data set (called a Repository Data Set or RDS). StarTool DA accesses the Repository Data Set for source level debugging with the Mini Dump or the Debug/Viewing Server (DVS).

You must also add the following DD statement to the JCL that executes the program:

```
//ESPYCOBL DD DSN=xxxxxxxx
```

where xxxxxxxx is the name of the sequential or partitioned data set that contains the compiled source listing.

Development Mode supports COBOL II, LE/370 (SAA AD/CYCLE COBOL/370), LE/390, COBOL for MVS, and the old VS-COBOL.



NOTE After-the-abend compiles are not supported in Development Mode.

Differences Between Production and Development

Processing differs between Production Mode and Development Mode at abend time and when using DVS.

At abend time

When Workload Server processing searches for the COBOL source, the following occurs:

- **For a production abend . . .**

The Language Server searches the language database for the matching source, verifies security access, and extracts the source information. If the security check fails, no language database processing occurs. The Language Server provides a return code to the Workload Server and processing continues as if this were a development abend.

If the source is found, and a DVS dump is also being taken, then StarTool DA Batch writes the source to the DataBase Index (DBI). When you view the dump, the system need not search for the source.

In some cases, StarTool DA Batch calls the Language Server again within DVS. Then it places a flag in the DBI header record to identify the type of dump (production or development).

- **For a development abend . . .**

If the execution JCL contains the ESPYCOBL DD statement and the EMCT (MVS Control Table) indicates that a Mini Dump is to be taken, then StarTool DA uses the ESPYCOBL data set as the compile listing repository.

When Using DVS

When you select a DVS dump for viewing and StarTool DA Batch is ready to access the source, the following occurs:

- **For a production dump . . .**

If the source was saved in the DataBase Index as a subset of the header records, then that source is used; otherwise, the Language Server extracts the source information from the language database and writes it to the DataBase Index.

- **For a development dump . . .**

If the source was saved in the DataBase Index as a subset of the header records, then that source is used; otherwise, the repository data set referenced by the ESPYCOBL DD is used. If the source is found, it is then written to the DataBase Index.

Language Environment

StarTool DA Batch can diagnose errors that occur in COBOL programs running under IBM's Language Environment (LE). LE can be configured to trap application errors such that the normal operating system abend and dump services do not receive control. LE thus provides its own condition-handling and error reporting procedures. Part of the condition-handling that LE performs is processing to allow registered Abnormal Termination Exits (ATEs) to receive control in the event of an error.

Invoking StarTool DA-Batch within IBM's Language Environment

Since the StarTool DA install registers the StarTool DA ATE (Abnormal Termination Exit) with LE, StarTool DA receives control from LE for any unhandled condition of severity 2 or greater.

After StarTool DA is installed, COBOL application programs are automatically processed by StarTool DA during abnormal termination processing.



NOTE The LE (Language Environment) installation job must be run when LE maintenance is applied or LE is upgraded.

The installation procedure does not use SMP/E, so you need to perform a manual link-edit of the necessary load modules. The job must be rerun when LE Maintenance is applied or LE is upgraded. Make a backup copy of your production SCEERUN (run-time) library before running the job.

Debugging and Viewing Server Call/Trace Summary for COBOL Programs

The Debugging and Viewing Server (DVS) supports interactive viewing of program working storage and data storage for selected COBOL application programs in the calling chain. This feature is invoked from the **Program Call/Trace Summary** panel in DVS.

```
T----- StarTool DA Program Call / Trace Summary          Row 1 to 4 of 4
COMMAND ==> _____          SCROLL ==> CSR

          Call Trace Summary and Application Program Attributes

To display a called program's COBOL storage and data, enter "S" into the
selection column and press enter. (Press PF1 for additional help.)

---Called Pgm---  ---Called Pgm Attributes---  -----Caller-----  ---Call----
S Program  Load-Mod  Comp Date  Length  Language  Load-Mod Program  Offset Mode
-----
_ ESPYCB39  ESPYCB39  04 Aug 2005  002B14  COBOL/ZOS  *SYSTEM*          CALL
_ ESPYCB39  ESPYCB39          CEEBINIT CEEBTOR  000000 CALL
_ IGZCFCC  IGZCPAC   30 Jun 2003  000E00  HLA ASMBLR  ESPYCB39  ESPYCB39  0017EA CALL
_ ESPYCB3B  ESPYCB3B  04 Aug 2005  000674  COBOL/ZOS  IGZCPAC  IGZCFCC  0002CC CALL
***** Bottom of data *****
```



NOTE This panel lists *called* programs on the left and their *calling* programs on the right.

Only called programs that have completed COBOL initialization processing are eligible for viewing. These programs are marked in the panel with an underscore in the **S** column on the far left. Enter an 'S' line command to select the desired called program for viewing.

The **DVS Selection Menu for COBOL Programs** panel (ESPYIPAI) displays. The name of the chosen program appears in the upper left.

```
ESPYIPAI ----- StarTool DA : DVS Selection Menu for COBOL Programs
COMMAND/OPTION ==>

COBOL program name : ESPYCB39          Compile time/date : 2005 Aug 04 10:29:31.98
                                COBOL compiler   : COBOL/ZOS

          COBOL program storage display options
          1 Display Linkage-Section (BLL) storage
          2 Display Working-Storage (BLW) storage
          3 Display program storage

          5 Display program source listing

          COBOL control block display options
          4 Display COBOL TGT
          5 Display COBOL RunCom
          6 Display COBOL DSA

Enter number of selected option or END(PF3) to exit.
```

The panel allows you to view COBOL Linkage-Section storage areas, Working-Storage-Section storage areas, program storage, the program source code listing, or various COBOL control blocks associated with the selected program. Type the desired option

number at the **COMMAND/OPTION ===>** prompt and press Enter. A list of all storage areas of the selected type is displayed.

For example, if you enter Option 1 on the preceding panel, DVS displays the **COBOL BLL Cell Storage Display** panel (ESPYIPAJ) for your chosen program. All BLL cells associated with the program are listed in display.

```

ESPYIPAJ ----- StarTool DA-Batch COBOL BLL Cell Storage Display
COMMAND ===> _____ SCROLL ===> CSR

Current COBOL program : ESPYCB39

Input   Cell       Address      Length      Description
-       00000001    00000000    00000000    -- Storage address is zero
-       00000002    1950D478    00000004
-       00000003    00000000    00000000    -- Storage address is zero
***** Bottom of data *****
    
```

To view the contents of a particular storage cell, type an 'S' in the **Input** column to the left of the desired cell number and press Enter. The contents of that cell displays in the DVS **Storage Display** panel.

Chapter 5

DB2 Option

The DB2 option is a separately licensed feature of StarTool DA. It provides access to DB2 related diagnostics information for call-attach programs, TSO-attach programs, and stored procedures.

DB2 Diagnostics	76
SQL Call Summary Panel	76
Plan Information Panel	77
Package Information Panel	80
DBRM Information Panel	84
SQL Statement and Host Variables Panel	86

DB2 Diagnostics

You can view the diagnostics in the Mini Dump report or you can select Option 9, DB2 SQL Information, on the DVS Selection Menu.

```
----- StarTool DA : DVS Selection Menu -----  
COMMAND ==>>  
  
Title: ABEND S0C7 IN JOB ES992ACC AT 11:36:21 ON SATURDAY, March 15, 2007  
  
          1 System and Job Information  
          2 Diagnostic Section  
          3 Failing Instruction Section  
          4 Reg Contents and PSW Info  
          5 Pgm Call Trace Summary  
          6 Program Storage  
          7 Pgm Listing (COBOL Only)  
          8 Data Mgmt Control Blocks  
          C ChangeMan ZMF Information  
          D Program Call/Trace Details  
          L Load a source listing to DA  
  
Enter Number of selected Option or END(PF3) To EXIT.
```

Abnormal termination of an application occurs as a result of a DB2 SQL call (EXEC SQL) or it can follow an SQL call. In either case, the StarTool DA DB2 Interface (hereafter referred to as the DB2 Interface), references the DB2 application environment at the time of the last SQL call.

To provide access to DB2 call parameters following an unsuccessful SQL call, the application abnormally terminates with a user abend. Use the EXEC SQL WHENEVER clause to do this. A call to module ILBOABN0 when an SQL error is encountered provides a user abend.

The following logic flow summary describes only general flow, is provided for general information purposes only, is *not* provided as program or product documentation, and may be altered or changed at any time without a product update.

SQL Call Summary Panel

In this example, an application is issuing a user abend in response to a negative SQL return code.

```

----- StarTool DA : Job SQL Listing -----
COMMAND ==>>>                                SCROLL ==>> PAGE

Selected Job= ES992ACC
Program=      ESPUSACC

----- S Q L   S T A T E M E N T S -----
SQL Call Summary

DB2 Subsystem Name(Q101)           Plan Name(ESPYD2PL)
Program Name(ESPUSACC)            Statement Number(204)
SQL Timestamp(16DEFEE0069ACD4A)   Address of SQL ParmList(00008CE8

DSNT400I  SQLCODE = -818, ERROR:  THE PRECOMPILER-GENERATED TIMESTAMP
        169C2A0B06748E2C IN THE LOAD MODULE IS DIFFERENT FROM THE BIND
        TIMESTAMP 169C296E09A2D6A6 BUILT FROM THE DBRM ESPUSACC
DSNT418I  SQLSTATE  = 51003 SQLSTATE RETURN CODE
DSNT418I  SQLERRP   = DSNXEPM SQL PROCEDURE DETECTING ERROR
DSNT418I  SQLERRD   = -200 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT418I  SQLERRD   = X'FFFFFF38' X'00000000' X'00000000'
        X'FFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC INFORMATION

```

Field Definitions	
DB2 Subsystem Name	JES Subsystem Vector Table Entry name for DB2 subsystem at time of transaction abend.
Plan Name	Identification of the DB2 plan bound to the application program in control at time of transaction abend.
Program Name	Identification of the DBRM library member that contains the statement associated with the last SQL call at time of transaction abend.
Statement Number	Identification of the SQL statement associated with the last SQL call at time of transaction abend.
SQL Timestamp	Identification of the iteration of the DBRM library member that contains the statement associated with the last SQL call at time of transaction abend.
Address of SQL Parmlist	The last SQL call parameter list passed to the DB2 Resource Manager at time of transaction abend.
SQL Communication Area	Formatted display of the SQLCA provided by the DB2 module DSNTIAR. If the DB2 Resource Manager Interface abnormally terminated during processing of an SQL call, a warning message displays because the SQLCA is not available.

Plan Information Panel

The Plan Information follows the SQL Call Summary section.

```

----- StarTool DA : Job SQL Listing -----
COMMAND ===>                                SCROLL ===> PAGE

Selected Job= ES992ACC
Program=      ESPUSACC

----- S Q L   S T A T E M E N T S -----

Plan Information

Plan(ESPYD2PL)                               Creator(SPANDA)
Bound by(SPANDA)                             Qualifier(SPANDA)
Bind Date(2002-03-15)                       Bind Time(11.58.22.100093)
Validate(Run Time)                          Isolation(Cursor Stability)
Valid(Yes)                                  Operative(Yes)
Acquire(Use)                                Release(Commit)
Current Data(Yes)                           Defer Prepare(No)
Current Server()                             Degree(1)
SQL Rules(DB2)                               Disconnect(Explicit)
Group Member()                               Dynamic Rules(Run)
Optimize Variable Input Values(No)          Keep Dynamic(NO)

```

Field Definitions	
Plan - NAME	Name of the application plan.
Creator - CREATOR	Authorization ID of the owner of the table, view, or alias.
Bound by - BOUNDBY	Primary authorization ID of the binder of the plan.
Qualifier - QUALIFIER	Implicit qualifier for the unqualified table, view, index, and alias names in the static SQL statements of the plan.
Bind Date - BOUNDTS	The date on which the plan was last bound, in the form yymmdd.
Bind Time - BOUNDTS	Time of the BIND in the form hhmmssst.
Validate - VALIDATE	Whether or not validity checking can be deferred until run time: <ul style="list-style-type: none"> ■ B - All checking must be performed during BIND. ■ R - Validation is done at run time for tables, views, and privileges that do not exist at bind time.
Operative - OPERATIVE	Whether or not the application plan can be allocated: <ul style="list-style-type: none"> ■ N - No; an explicit BIND or REBIND is required before the plan can be allocated. ■ Y - Yes.
Acquire - ACQUIRE	When resources are acquired: <ul style="list-style-type: none"> ■ A - At allocation. ■ U - At first use.

Field Definitions	
Release - RELEASE	When resources are released: <ul style="list-style-type: none"> ■ C - At commit. ■ D - At deallocation.
Current Data - EXPREDICATE	Indicates the CURRENTDATA option when the plan was bound or rebound: <ul style="list-style-type: none"> ■ B - Data currency is not required for ambiguous cursors. Allow blocking for ambiguous cursors. ■ C - Data currency is required for ambiguous cursors. Inhibit blocking for ambiguous cursors. ■ N - Blocking is inhibited for ambiguous cursors, but the plan was created before the CURRENTDATA option was available.
Defer Prepare - DEFERPREP	Whether or not the package was last bound with the DEFER(PREPARE) option: <ul style="list-style-type: none"> ■ N - No ■ Y - Yes
Current Server - CURRENTSERVER	Location name specified with the CURRENTSERVER option when the plan was last bound. Blank if none was specified, implying that the first server is the local DB2 subsystem.
Degree - DEGREE	The DEGREE option used when the plan was last bound: <ul style="list-style-type: none"> ■ ANY - DEGREE(ANY) ■ 1 or blank - DEGREE(1). Blank if the plan was migrated.
SQL Rules - SQLRULES	The SQLRULES option used when the plan was last bound: <ul style="list-style-type: none"> ■ D or blank - SQLRULES(DB2). ■ S - SQLRULES(STD). ■ blank - A migrated plan.
Disconnect - DISCONNECT	The DISCONNECT option used when the plan was last bound: <ul style="list-style-type: none"> ■ E or blank - DISCONNECT - (EXPLICIT). ■ A - DISCONNECT - (AUTOMATIC). ■ C - DISCONNECT - (CONDITIONAL). ■ blank - a migrated plan.
Group Member - GROUP_MEMBER	The DB2 data sharing member name of the DB2 subsystem that performed the most recent bind. This column is blank if the DB2 subsystem was not in a DB2 data sharing environment when the bind was performed.
Dynamic Rules - DYNAMICRULES	<ul style="list-style-type: none"> ■ B - Dynamic SQL statements are handled like static SQL statements at run time. ■ blank - Dynamic SQL statements are handled like dynamic SQL statements at run time.

Field Definitions	
Optimize Variable Input Values - REOPTVAR	Whether or not the access path is determined again at execution time using input variable values.
Keep Dynamic - KEEPDPYDYNAMIC	Whether or not prepared dynamic statements are to be purged at each commit point: <ul style="list-style-type: none"> ■ N - The bind option is KEEPDPYDYNAMIC(NO). Prepared dynamic SQL statements are destroyed at commit or rollback. ■ Y - The bind option is KEEPDPYDYNAMIC(YES). Prepared dynamic SQL statements are kept past commit or rollback.

Package Information Panel

Package Information is available in the StarTool DA dump only if the DB2 plan used by the abending program was bound with the PKLIST option.

```

----- StarTool DA : Job SQL Listing -----
COMMAND ==>>>                                SCROLL ==>>> CSR

Selected Job= ES992ACC
Program=      ESPUSACC

----- S Q L   S T A T E M E N T S -----
Package Information

Package(ESPUSACC)           Collection Id(ESPYD2)
Creator(SPANDA)             Owner(SPANDA)
Creation Date(2007-03-12)   Creation Time(14.17.16.172136)
Bind Date(2007-03-15)      Bind Time(10.06.04.147847)
Pre-compile Date(2007-03-10) Pre-compile Time(12.04.36.126517)
Qualifier(SPANDA)          Valid(Yes)
Operative(Yes)              Validate(Run Time)
Isolation(Cursor Stability) Release()
Quote(Apostrophe)           Comma(Period)
Host Language(IBM COBOL)    Katakana(No)
Mixed(No)                    Dec31(No)
Defer Prepare(Yes)          SQL Error(Nopackage)
Source(DBRM)                 Degree(1)
Version()
DBRM Library(SPANDA.DA.ES99.EDB2.DBRMLIB)
Group Member()              Dynamic Rules()
Optimize Variable Input Values(No) Keep Dynamic(NO)
Defer Prepare()

```

Field Definitions	
Package Name - NAME	Name of the package.
Collection Id - COLLID	Name of the package collection.

Field Definitions	
Creator - CREATOR	Authorization ID of the creator of the package version.
Owner - OWNER	Authorization ID of the package owner.
Creation Date - TIMESTAMP	Date/timestamp indicating when the package was created.
Creation Time - TIMESTAMP	Date/timestamp indicating when the package was created.
Pre-Compile Date - PCTIMESTAMP	Date (and time) the application program was precompiled, or 0001-01-01-00.00.00.000000 if the LEVEL precompiler option was used, or if the package came from a non-DB2 location.
Pre-Compile Time - PCTIMESTAMP	Time (and date) the application program was precompiled, or 0001-01-01-00.00.00.000000 if the LEVEL precompiler option was used, or if the package came from a non-DB2 location.
Qualifier - QUALIFIER	Implicit qualifier for the unqualified table, view, index, and alias names in the static SQL statements of the package.
Valid - VALID	Whether or not the package is valid: <ul style="list-style-type: none"> ■ A – The description of the table or base table of a view referenced by the package was changed by the ALTER TABLE statement. The change did not require the invalidation of the package. ■ H – The description of the table or base table of a view referenced by the package was changed by the ALTER TABLE statement. The change will invalidate the package for a DB2 release prior to Version 5. ■ N – No ■ Y – Yes
Operative - OPERATIVE	Whether or not the package can be allocated: <ul style="list-style-type: none"> ■ N – No; an explicit BIND or REBIND is required before the package can be allocated. ■ Y – Yes
Validate - VALIDATE	Whether or not validity checking can be deferred until run time: <ul style="list-style-type: none"> ■ B – All checking must be performed at bind time. ■ R – Validation is done at run time for tables, views, and privileges that do not exist at bind time.

Field Definitions	
Isolation - ISOLATION	<p>Isolation level when the package was last bound or rebound:</p> <ul style="list-style-type: none"> ■ R – RR (repeatable read) ■ T – RS (read stability) ■ S – CS (cursor stability) ■ U – UR (uncommitted read) ■ blank – Not specified; at the level specified for the plan executing the package
Comma - COMMA	<p>Decimal point representation for SQL statements in package:</p> <ul style="list-style-type: none"> ■ N – Period ■ Y – Comma
Quote - QUOTE	<p>SQL string delimiter for SQL statements in the package:</p> <ul style="list-style-type: none"> ■ N – Apostrophe ■ Y – Quotation mark
Host Language - HOSTLANG	<p>Host language for the package's DBRM:</p> <ul style="list-style-type: none"> ■ B – Package Information is available in the StarTool DA dump only if the DB2 plan used by the abending program was bound with the PKLIST option Assembler language. ■ C – OS/VS COBOL ■ D – C ■ F – FORTRAN ■ P – PL/I ■ 2 – VS COBOL II or IBM COBOL Release 1 (formerly called COBOL/370) ■ 3 – IBM COBOL (Release 2 or subsequent releases) ■ 4 – C++ ■ blank – For remotely bound packages
Mixed - MIXED	<p>Indicates whether or not mixed data was in effect when the package's program was precompiled:</p> <ul style="list-style-type: none"> ■ N – No ■ Y – YES
Dec31 - DEC31	<p>Indicates whether or not DEC31 was in effect when the package's program was precompiled (for more on when DEC31 is in effect, see "Arithmetic with Two Decimal Operands" in topic 3.14.6 in the DB2 SQL Reference):</p> <ul style="list-style-type: none"> ■ N – No ■ Y – YES

Field Definitions	
Defer Prepare - DEFERPREP	<p>Indicates the CURRENTDATA option when the package was bound or rebound:</p> <ul style="list-style-type: none"> ■ A – Data currency is required for all cursors. Inhibit blocking for all cursors. ■ B – Data currency is not required for ambiguous cursors. ■ C – Data currency is required for ambiguous cursors. ■ blank – The package was created before the CURRENT DATA option was available.
Degree - DEGREE	<p>The DEGREE option used when the package was last bound:</p> <ul style="list-style-type: none"> ■ ANY – DEGREE(ANY) ■ 1 or blank – EGREE(1). Blank if the package was migrated.
Version - VERSION	Version identifier for the package
DBRM Library - PDSNAME	<p>For a locally bound package, the name of the PDS (library) in which the package's DBRM is a member. For a locally copied package, the value in SYSPACKAGE.PDSNAME for the source package. Otherwise, the product signature of the bind requester followed by one of the following:</p> <ul style="list-style-type: none"> ■ The requester's location name if the product is DB2. ■ Otherwise, the requester's LU name enclosed in angle brackets; for example, <LUSQLDS>.
Group Member - GROUP_MEMBER	<p>The DB2 data sharing member name of the DB2 subsystem that performed the most recent bind. This column is blank if the DB2 subsystem was not in a DB2 data sharing environment when the bind was performed.</p>
Dynamic Rules - DYNAMICRULES	<ul style="list-style-type: none"> ■ B – Dynamic SQL statements are handled like static SQL statements at run time. ■ R – Dynamic SQL statements are handled like dynamic SQL statements at run time. ■ blank – DYNAMICRULES is not specified for the package. The package uses the DYNAMICRULES value of the plan to which the package is appended at execution time.
Optimize Input Variable Values - REOPTVAR	<p>Whether or not the access path is determined again at execution time using input variable values:</p> <ul style="list-style-type: none"> ■ N – Bind option NOREOPT(VARS) indicates that the access path is determined at bind time. ■ Y – Bind option REOPT(VARS) indicates that the access path is determined at execution time for SQL statements with variable values.

Field Definitions	
Keep Dynamic - KEEPDYNAMIC	Whether or not prepared dynamic statements are to be purged at each commit point: <ul style="list-style-type: none"> ■ N – The bind option is KEEPDYNAMIC(NO). Prepared dynamic SQL statements are destroyed at commit. ■ Y – The bind option is KEEPDYNAMIC(YES). Prepared dynamic SQL statements are kept past commit.
Defer Prepare - DEFERPREPARE	Whether or not PREPARE processing is deferred until OPEN is executed: <ul style="list-style-type: none"> ■ N – Bind option NODEFER(PREPARE) indicates that PREPARE processing is not deferred until OPEN is executed. ■ Y – Bind option DEFER(PREPARE) indicates that PREPARE processing is deferred until OPEN is executed. ■ blank – Bind option not specified for the package. It is inherited from the plan.

DBRM Information Panel



NOTE DBRM Information is available in the StarTool DA dump only if the DB2 plan used by the abending program was bound with the MEMBER option.

```

----- StarTool DA : Job SQL Listing -----
COMMAND ==>>>                                SCROLL ==>> CSR

Selected Job= ES992ACC
Program=      ESPUSACC

----- S Q L   S T A T E M E N T S -----
  DBRM Information

  DBRM(ESPUSACC)                               Creator(WST004)
  Pre-compile Date(2007-03-10)                 Pre-compile Time(12.04.36.126517)
  Quote(Apostrophe)                           Comma(Period)
  Host Language(IBM COBOL)                     Katakana(No)
  Mixed(No)                                    Dec31(No)
  Version()
  DBRM Library(SPANDA.DA.ES99.EDB2.DBRMLIB)

```

Field Definitions	
DBRM - NAME	Name of the DBRM.
Creator - PLCREATOR	Authorization ID of the owner of the application plan.

Field Definitions	
Pre-Compile Date - PRECOMPTIME	Time of precompilation in the form <i>hhmmssth</i> . If the LEVEL precompiler option is used, then this value does not represent the precompile time.
Pre-Compile Time - PRECOMPDATE	Date of precompilation in the form <i>yyymmdd</i> . If the LEVEL precompiler option is used, then this value does not represent the precompile date.
Quote - QUOTE	SQL string delimiter for the SQL statements in the DBRM: <ul style="list-style-type: none"> ■ N - Apostrophe ■ Y - Quotation mark
Comma - COMMA	Decimal point representation for SQL statements in the DBRM: <ul style="list-style-type: none"> ■ N - period ■ Y - comma
Host Language - HOSTLANG	The host language used: <ul style="list-style-type: none"> ■ B - assembler language ■ C - COBOL ■ D - C ■ F - FORTRAN ■ P - PL/I ■ 2 - VS COBOL II
Katakana - CHARSET	Indicates whether not the system CCSID for SBCS data was 290 (Katakana) when the program was precompiled: <ul style="list-style-type: none"> ■ A - No ■ K - Yes
Mixed - MIXED	Indicates if mixed data was in effect when the application program was precompiled (for more on when mixed data is in effect, see " <i>Character Strings</i> " in topic 3.7.1 in the DB2 SQL Reference): <ul style="list-style-type: none"> ■ N - No ■ Y - Yes
Dec31 - DEC31	Indicates whether or not DEC31 was in effect when the program was precompiled (for more on when DEC31 is in effect, see " <i>Arithmetic with Two Decimal Operands</i> " in topic 3.14.6 in the DB2 SQL Reference): <ul style="list-style-type: none"> ■ blank - No ■ Y - Yes
Version - VERSION	Version identifier for the DBRM.
DBRM Library - PDSNAME	Name of the partitioned data set of which the DBRM is a member.

SQL Statement and Host Variables Panel

The SQL Statement section displays the last SQL Statement issued by the application program.

```
----- StarTool DA : Job SQL Listing -----
COMMAND ==>                                     SCROLL ==> CSR

Selected Job= ES992ACC
Program=      ESPUSACC

----- S Q L   S T A T E M E N T S -----
SQL Statement

SELECT
  COLLID ,
  NAME ,
  OWNER ,
  CREATOR ,
  TIMESTAMP ,
  BINDTIME ,
  QUALIFIER
INTO
  : WS-SYSPACKAGE.WS-SYSPACKAGE-COLLID ,
  : WS-SYSPACKAGE.WS-SYSPACKAGE-NAME ,
  : WS-SYSPACKAGE.WS-SYSPACKAGE-OWNER ,
  : WS-SYSPACKAGE.WS-SYSPACKAGE-CREATOR ,
  : WS-SYSPACKAGE.WS-SYSPACKAGE-TIMESTAMP ,
  : WS-SYSPACKAGE.WS-SYSPACKAGE-BINDTIME ,
  : WS-SYSPACKAGE.WS-SYSPACKAGE-QUALIFIER
FROM
  SYSIBM . SYSPACKAGE
WHERE COLLID = : WS-SYSPACKAGE.WS-SYSPACKAGE-COLLID
AND NAME = : WS-SYSPACKAGE.WS-SYSPACKAGE-NAME
```

The Host Variables section contains the attributes of the host variables associated with the last SQL call at time of transaction abend.

Host Variables		
Name (WS-SYSPACKAGE.WS-SYSPACKAGE-COLLID)		
Type (CHAR)	Length (18)	Address (000085F8)
Value (ESPYD2)		
Name (WS-SYSPACKAGE.WS-SYSPACKAGE-NAME)		
Type (CHAR)	Length (8)	Address (0000860A)
Value (ESPUSACC)		
Name (WS-SYSPACKAGE.WS-SYSPACKAGE-OWNER)		
Type (CHAR)	Length (8)	Address (0000861A)
Value (SPANDA)		
Name (WS-SYSPACKAGE.WS-SYSPACKAGE-CREATOR)		
Type (CHAR)	Length (8)	Address (00008622)
Value (SPANDA)		
Name (WS-SYSPACKAGE.WS-SYSPACKAGE-TIMESTAMP)		
Type (CHAR)	Length (26)	Address (0000862A)
Value (2007-01-17-14.17.16.172136)		
Name (WS-SYSPACKAGE.WS-SYSPACKAGE-BINDTIME)		
Type (CHAR)	Length (26)	Address (00008644)
Value (2007-01-03-10.06.04.147847)		
Name (WS-SYSPACKAGE.WS-SYSPACKAGE-QUALIFIER)		
Type (CHAR)	Length (8)	Address (0000865E)
Value (SPANDA)		

Field Definitions	
Name	Name of host variable.
Type	Type of host variable: character, packed, or binary.
Length	Length of host variable.
Address	Address of host variable.
Value	Value of host variable. For character type host variables, a maximum of seventy-two characters display. Using the Point-and-Shoot feature on the host variable address, the complete value displays.

Chapter 6

IMS Option

The IMS option is a separately licensed feature of StarTool DA. Use this option to analyze IMS application program dumps. It provides IMS related diagnostic information that you can view in the IMS Data section of the Mini Dump report or by choosing Option 10 from the DVS Selection Menu; both ways provide the same information.

What is Supported	90
Information Displayed	91
Viewing a Dump with DVS	92

What is Supported

The following table shows what region types, PCBs, PSBGEN languages, and DL/I call languages are supported.

For this category	This is supported
Region types	<ul style="list-style-type: none"> ■ DL/I Batch ■ BMP ■ MPP ■ IFP (Fast Path) ■ DBB
PCBs	<ul style="list-style-type: none"> ■ TP (I/O and Alternate) ■ DB ■ Fast Path (MSDB & DEDB) ■ GSAM
PSBGEN languages	<ul style="list-style-type: none"> ■ ASSEMBLER ■ COBOL ■ FORTRAN ■ PASCAL ■ PL/I
DL/I-Call language interfaces	<ul style="list-style-type: none"> ■ ASSEMBLER ■ COBOL ■ FORTRAN ■ PASCAL ■ PL/I ■ AIB

Information Displayed

The following table shows what information displays.

For this category	The following information displays
DLI, BMP, MPP, IFP, and DBB region types	<ul style="list-style-type: none"> ■ Job Name ■ Step Name ■ IMS Release ■ IMS Subsystem Name ■ Region Type ■ PSB Name ■ PSBGEN Language Option ■ Last DL/I-Call Language Interface ■ Application Program Name ■ Application Program Input Parameter List
TP, DB, and GSAM PCBs	<ul style="list-style-type: none"> ■ PCB Address ■ PCB Type ■ PCB Name ■ PCB Relative Number ■ Relocated PCB Address (for BMP region type) ■ Indication of last DL/I call using this PCB ■ Indication of LIST=NO option in PCBGEN/PSBGEN (for BMP region type) ■ Status Code ■ Status Code Explanation for a non-blank status code ■ Application Programmer Response for a non-blank status code
TP PCB	<ul style="list-style-type: none"> ■ Terminal Symbolic Name ■ Default Formatting Module Name ■ User ID ■ Group Name

For this category	The following information displays
DB PCB	<ul style="list-style-type: none"> ■ DBD Name ■ Processing Option ■ Number of Sensitive Segments ■ Level Feedback ■ Segment Name Feedback ■ Key Feedback Area ■ Segment Data (for DL/I Batch and DBB region types) ■ JCB DL/I-Call Trace Entries in FIFO (First-In-First-Out) order (for DL/I Batch and DBB region types)
GSAM PCB	<ul style="list-style-type: none"> ■ DBD Name ■ Processing Option ■ DDNAME ■ Data Set Organization ■ Data Set Name(s) & Volume Serial Number(s) ■ User I/O Area
Last DL/I-Call PCB	<ul style="list-style-type: none"> ■ Last DL/I-Call Language Interface ■ Last DL/I-Call User ParmList address, module name, and offset ■ Last DL/I-Call User Parm addresses, module names, and offsets ■ Last DL/I-Call Function Code and its address



NOTE Any reference to language support for the IMS Option is limited to displaying information such as control block contents, parameter addresses, parameter values, and so on.

There is no point-and-shoot support for high-level language listings in the IMS Option; point-and-shoot is available only for storage addresses in the dump.

Viewing a Dump with DVS

This section describes the features and diagnostic information available for an IMS DVS dump.



NOTE The information available with DVS is the same as the IMS Data section of the Mini Dump report. With DVS, you can use the Point-and-Shoot feature to display storage locations.

To access IMS dump information using DVS, choose Option 10 from the DVS Selection Menu.

```

----- StarTool DA : DVS Selection Menu -----
COMMAND ==>>

Title: ABEND S0C1  IN JOB TEST1234 AT 15:11:22 ON SATURDAY, March 15, 2007

          1  System and Job Information
          2  Diagnostic Section
          3  Failing Instruction Section
          4  Reg Contents and PSW Info
          5  Pgm Call Trace Summary
          6  Program Storage
          7  Pgm Listing (COBOL Only)
          8  Data Mgmt Control Blocks
         10  IMS Information
          C  ChangeMan ZMF Information
          D  Program Call/Trace Details
          L  Load a source listing to DA

Enter Number of selected Option or END(PF3) To EXIT.

```



NOTE Option 10 may not be on the menu if IMS primary modules (such as DFSRRC00 and DFSVC000) are not present in the dump or are not fully initialized when the error occurs.

This is an example of the first screen to appear for a DL/I dump.

```

----- StarTool DA : Job IMS Listing -----
COMMAND ==>>                                SCROLL ==>> PAGE

Selected Job= TEST1234
Program=      IMSLINK9

----- I M S   D A T A -----

Job Name = TEST1234
Step Name = DLI6
IMS Release = 06.1.0
IMS Subsystem Name = IVP1
IMS Region Type = DL/I - DL/I Batch Region

PSB Name = ATAMDLI6
PSBGEN Language Option = ASSEM/COBOL/FORTRAN/blank or default
Last DL/I-Call Language Interface = Assembler Call
Appl Program Name = IMSDLI6

Appl Pgm Input ParmList at x'00005FE0' (20 Bytes):
-----1-----2-----3-----4-----5-----6-----7
DLI,IMSDLI6,ATAMDLI6

```

Subsequent screens contain information for each PCB. The following screen shows the GSAM PCB, which was the PCB used by the last DL/I call. It contains the Last DL/I-Call information, including the addresses of the User ParmList and Function Code.

```

----- StarTool DA : Job IMS Listing -----
COMMAND ==>                                SCROLL ==> PAGE

Selected Job= TEST1234
Program=     IMSLINK9

----- I M S   D A T A -----

***** GSAM PCB at x'800AF2E8' (Last DL/I Call) *****

PCB Name = GSAMPCB
PCB Relative Number = 5
DBD Name = ATAMDBD1
Processing Option = G
Status Code = x'4040' c'  '

DDNAME = ATAMDD0
Data Set Organization = BSAM
Data Set Names (Volume Serial Numbers):
ATAM.DBDGSAM0 (SMS001)
ATAM.DBDGSAM1 (SMS005)

User I/O Area at x'0D861650' (80 Bytes):
----+----1----+----2----+----3----+----4----+----5----+----6----+----7
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aa00000100

Last DL/I-Call Language Interface = Assembler Call
Last DL/I-Call User ParmList at x'0D8613EC' (IMSDLI6 + x'0001E4'):
    0D861448 000AF2E8 8D861650
Last DL/I-Call Function Code at x'0D861448' = x'C7D54040' c'GN  '

    Parm at x'0D861448' ==> IMSDLI6 + x'000240'
    Parm at x'000AF2E8' ==> Addr doesn't point to an active load module.
    Parm at x'8D861650' ==> IMSDLI6 + x'000448'

***** End of IMS Data Section *****

***** Bottom of data *****

```



NOTE In the Last DL/I-Call Language Interface field, the value displays as follows for these conditions:

- When using the Language Environment interface, the value is Unknown.
- When using the C language interface, the value is Assembler.

Point-and-Shoot

Use Point-and-Shoot with the IMS Option to display storage locations. To use this feature, place the cursor on any line with an 8-byte hexadecimal address preceded by 'at' or 'to', and press Enter.

The following example shows information for the I/O PCB. To get to the storage location, place the cursor anywhere on the line containing the statement I/O PCB at x'000310BC', and press Enter.

```

----- StarTool DA : Job IMS Listing -----
COMMAND ==>                                SCROLL ==> PAGE

Selected Job= TEST1234
Program=     IMSLINK9

----- I M S   D A T A -----

***** I/O  PCB at x'000310BC' *****

PCB Name = IOPCB
PCB Relative Number = 1

Status Code = x'C1D3' c'AL'

Status Code Explanation:
Invalid call to IOPCB in batch DL/I.

Appl Programmer Response:
Probable programming error.
Batch DL/I cannot issue calls to the message queue.
In batch DL/I, ROL/ROLB calls are valid only if BKO was
specified the DASD logs are being used. Correct program.

Terminal Symbolic Name = PSTVALID
Default Formatting Module Name =
User ID =
Group Name =

```

The Storage Display screen displays for the I/O PCB.

```

----- StarTool DA : Storage Display -----
COMMAND ==>                                SCROLL ==> PAGE

Selected Job= TEST1234
Start Address= 000310BC  Area= PVT   Program=

Address  Offset      Data (Hexadecimal)      Data (EBCDIC)
000310BC 000000  D7E2E3E5  C1D3C9C4  1000C1D3  40404040  * PSTVALID. AL  *
000310CC 000010  40404040  40404040  40404040  40404040  *                *
000310DC 000020  40404040  40404040  40404040  40404040  *                *
000310EC 000030  00000000  00000000  00000000  00000000  *                *
000310FC 000040  00000000  00000000  00000000  00000000  *                *
0003110C 000050  00000000  00000000  00000000  00000000  *                *
0003111C 000060  00000000  00000000  00000000  00000000  *                *
0003112C 000070  00000000  00000000  00000000  00000000  *                *
0003113C 000080  00000000  00000000  00009D58  00009D58  *                *
0003114C 000090  008F62C0  0D8615D0  00000050  0D8615D0  * ±Å{.f.}  &.f.} *
0003115C 0000A0  0003A544  00000000  00000000  00000000  * .vâ      *
0003116C 0000B0  00000000  00000000  000072D0  00000000  *                Ê}  *

```



NOTE When you go to the Storage Display screen using Point-and-Shoot, you cannot scroll backwards to an address prior to the Start Address. To go to a prior address, type the following command on the command line:

```
L xxxxxxxx
```

where xxxxxxxx is a 1- to 8-byte hexadecimal address.

Relocated PCBs

The following example shows a PCB that was relocated by IMS. If you place the cursor on the line containing the statement `DB PCB at x'0000DB24'` and press Enter, you will get an error message indicating that the storage location is not in the dump, or find that the location exists but has unexpected data. This can happen if IMS releases the control block and the storage is then allocated to another program. To get to the relocated storage location, place the cursor anywhere on the line containing the statement `relocated to x'00015D6C'`, and press Enter.

```
----- StarTool DA : Job IMS Listing -----
COMMAND ==>                                SCROLL ==> PAGE

Selected Job= TEST1234
Program=     IMSLINK9

----- I M S   D A T A -----

***** DB PCB at x'0000DB24' *****

Warning Message(s):
This PCB has been relocated to x'00015D6C' by IMS.
PCB storage at x'0000DB24' may have been released by IMS,
and then allocated/reused by another system program.

PCB Name = DBPCB1
PCB Relative Number = 2
DBD Name = GEMSDB
Processing Option = AP
Status Code = x'4040' c' '

No. of Sensitive Segments = 12
Level Feedback = 02
Segment Name Feedback = COLOUR

Key Feedback Area at x'00015D90' (30 Bytes):
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7
RUBY                                     RED
***** DB PCB at x'0000DC08' *****
```


The Storage Display screen displays for the relocated PCB.

```

----- StarTool DA : Storage Display -----
COMMAND ==>                                SCROLL ==> PAGE

Selected Job= USER298B
Start Address= 00015D6C Area= PVT Program=

Address  Offset      Data (Hexadecimal)      Data (EBCDIC)
00015D6C 000000  C7C5D4E2  C4C24040  F0F24040  C1D74040  * GEMSDB 02 AP
00015D7C 000010  0000DAEC  C3D6D3D6  E4D94040  0000001E  * 'ÖCOLOUR
00015D8C 000020  0000000C  D9E4C2E8  40404040  40404040  * .RUBY
00015D9C 000030  40404040  40404040  D9C5C440  40404040  * RED
00015DAC 000040  40404040  40404040  40404040  40404040  *
00015DBC 000050  40404040  40404040  40404040  40404040  *
00015DCC 000060  40404040  40404040  40404040  40404040  *

```

IMS Region Abends

The following example shows an IMS region abend.

Dumps initiated by an IMS module (ABEND SVC-13) display the IMS User Abend Code at the beginning of the IMS Data section. Because there are so many IMS User Abend Codes, an explanatory message displays only for the most common ones. Complete information is available in the following IBM publications:

- *IMS/ESA Messages and Codes*
- *IMS/ESA Failure Analysis Structure Table for Dump Analysis*

```
----- StarTool DA : Job IMS Listing -----
COMMAND ==>                                     SCROLL ==> PAGE

----- I M S   D A T A -----

IMS Region ABEND U0456
ABEND SVC-13 was issued at x'0000879C' (DFSPCC20 + x'000DEC').

U0456 - BMP/IFP not initiated, PSB has been stopped or locked.

Refer to following IBM publications for complete information:

IMS/ESA Messages and Codes
IMS/ESA Failure Analysis Structure Table for Dump Analysis

Correct the original problem and resubmit the job.

Job Name = MYBMPJOB
Step Name = BMP10
IMS Release = 07.1.0
IMS Subsystem Name = D111

Unable to locate IMS Control Block DIRCA.
IMS Region may not be fully initialized in this address space.
Correct the original problem and resubmit the job.

Appl Pgm Input ParmList at x'00015D90' (34 Bytes):
-----1-----2-----3-----4-----5-----6-----7
BMP,IMSFP1A,DFSIVP6,,,,,,,,,D111

***** End of IMS Data Section *****
```

Chapter 7

ChangeMan ZMF Option

If StarTool DA intercepts an abending routine that is controlled by ChangeMan ZMF, StarTool DA accesses ChangeMan ZMF and retrieves data for the package that the abending job participates in.



NOTE Before you can use this feature, you need to configure ChangeMan ZMF and StarTool DA as described in the *StarTool DA Installation Guide*. Remember to initialize the Serena® StarTool® DA Workload Server (WLS) *after* starting TCP/IP and the ChangeMan ZMF subsystem to which you want to connect.

Selecting the Dump to Analyze	100
General Package Information Panel	101
Load Component Information Panel	102
History Information Panel	103
Display ChangeMan Source Listings	104

Selecting the Dump to Analyze

To analyze a dump for a abended routine that was controlled by ChangeMan ZMF:

- 1 Access the DVS Selection Menu panel.
- 2 Select a dump, then press Enter.

If the dump was a transaction dump, the DVS Selection Menu screen displays.

```

----- StarTool DA : DVS Selection Menu -----
COMMAND ==>>

Title: ABEND S0C1  IN JOB WSER2769 AT 10:09:01 ON TUESDAY  , JUNE      29, 2007

          1  System and Job Information
          2  Diagnostic Section
          3  Failing Instruction Section
          4  Reg Contents and PSW Info
          5  Pgm Call Trace Summary
          6  Program Storage
          7  Pgm Listing (COBOL Only)
          8  Data Mgmt Control Blocks
          C  ChangeMan ZMF Information
          D  Program Call/Trace Details
          L  Load a source listing to DA

          Enter number of selected Option or END(PF3) To EXIT.

```

- 3 Choose **Option C, ChangeMan ZMF Information** from the Master Menu to view ChangeMan ZMF data.

This displays the main ChangeMan ZMF Information screen, where you can choose from the following options:

- [General Package Information Panel](#)
- [Load Component Information Panel](#)
- [History Information Panel](#)
- [Display ChangeMan Source Listings](#)

- 4 Select one of the options by typing the option number in the command line and pressing Enter.

```

----- StarTool DA : ChangeMan ZMF Information -----
COMMAND ==>>

The following options are selected in sequence, or may be selected by number :

          1. General Package Information
          2. Load Component Information
          3. History Information

The following option is selectable only by number :

          S. Display Source Listing

```



TIP Cycle through the first three options by pressing Enter without selecting a line. Continue to press Enter to display the following option.

General Package Information Panel

The General Package Information panel displays information about the package that contains the component that abended.

To access the General Package Information screen, select option 1 on the main ChangeMan ZMF Information panel.

```

----- StarTool DA : ChangeMan ZMF Information -----
COMMAND ==>>

      PACKAGE ID: STDA000004      STATUS: DEV      INSTALL DATE: 20071215
COMPLEX/SUPER ID: N/A

PACKAGE TITLE
==>> RUDY&APOS;S TEST PACKAGE

APPLICATION          ==>> STDA
REQUESTER'S NAME     ==>> RUDY
REQUESTER'S PHONE    ==>> 6593
WORK REQUEST ID      ==>>
DEPARTMENT           ==>> QA
PACKAGE LEVEL        ==>> 1          (1-Simple, 2-Complex,
                                     (3-Super, 4-Participating)

PACKAGE TYPE          ==>> PLANNED
PACKAGE TIME SPAN    ==>> PERM
UNPLANNED REASON CODE ==>>
TEMPORARY CHANGE DURATION ==>>

```

This table describes the fields on the General Package Information panel.

Field	Description
Package Title	The title of the change package.
Application	The application associated with the change package.
Requester's Name	The name of the person who requested the information.
Requester's Phone	The telephone number, or extension, of the requester.
Work Request ID	The Work Request ID.
Department	The requester's department.

Field	Description
Package Level	<p>Can be one of the following:</p> <p>1 (Simple). The change package contains a change that is unrelated to any other change package. This level does not affect any other application, nor does it require changes to software or operational procedures.</p> <p>2 (Complex). The parent for two or more participating change packages that have interdependent changes to software or operational procedures. Complex change packages contain only control, general information, and a list of participating packages. There are no staging libraries associated with super/complex change packages. Remote sites and the installation dates for each site are tracked in the participating change packages.</p> <p>3 (Super). The parent for two or more participating change packages that contain major changes to several applications. Super change packages contain only control, general information, and a list of the participating packages. There are no staging libraries associated with super/complex change packages. Remote sites and the installation dates for each site are tracked in the participating change packages.</p> <p>4 (Participating). The change package is related to one or more other participating change packages.</p>
Package Type	<p>Planned – Indicates that the packages are scheduled changes that follow the established application rules.</p> <p>Unplanned – Indicates that the packages are unscheduled changes, possibly an emergency fix. Emergency packages bypass some of the application rules and have a separate approval list.</p>
Package Time Span	<p>Permanent – Indicates that the packages are scheduled changes that update baseline or production libraries. Permanent changes are either planned or unplanned. This is the normal way to install changes.</p> <p>Temporary – Changes that run for a predetermined length of time. Once that time expires, ChangeMan ZMF automatically deletes the change. Temporary changes do not update baseline or production libraries. These changes are installed into a static override library. Temporary changes are either planned or unplanned.</p>
Unplanned Reason Code	The reason for the package being unplanned.
Temporary Change Duration	The number of days that the change is to remain in effect.

Load Component Information Panel

The Load Component Information panel displays ChangeMan ZMF information about the component that abended, such as source module name and last time the code was altered.

To access the Load Component Information panel, select option 2 on the main ChangeMan ZMF Information panel.

```

----- StarTool DA : ChangeMan ZMF Information -----
COMMAND ==>>

      PACKAGE ID: STDA000004      STATUS: DEV      INSTALL DATE: 20071215
      SOURCE          LOAD
NAME   TYPE   NAME   TYPE STATUS      PROMOTION      CHANGED      ID
ESPYCOB3 SRC   ESPYCOB3 LOD  ACTIVE                20070618 091416 RFERNAN

```

This table describes the fields on the Load Component Information panel.

Package ID	Displays the package ID for the package that contains the component which abended.
Status	Displays the status of the package.
Install Date	Displays the installation date for this package.
Source Name/Type	Displays the name and type of the source for the component that abended.
Load Name/Type	Displays the name and type of the load module for the component that abended.
Status	Displays the status of the component.
Promotion	Displays the promotion level and nickname for the promotion site.
Changed	Displays the date and time of the last edit to change to the component.
ID	Displays the TSO ID of the last user to alter the component.

History Information Panel

The History Information panel displays ChangeMan ZMF history information about the component that abended, such as version number and size.

To access the History Information panel, select option 3 on the main ChangeMan ZMF Information panel.

```

----- StarTool DA : ChangeMan ZMF HISTORY Information -----
COMMAND ==>
HISTORY Information:
PACKAGE ID STA PROMOTION VV.MM LAST ACTION SIZE PROCNAME ID
DA54000015 BAS 01.04 2003/03/17 05:59 00757 DA5COB2 WSER78
DA55000003 BAS 64.73 2005/08/03 11:50 00771 DA5COB2 WSER04
DA54000034 DEV 01.04 2004/03/24 08:31 00790 DA5COB2 WSER78
DA54000037 BAS 01.00 2004/06/09 21:53 UNKN DA5COB2 WSER78
DA55000001 BAS 01.00 2005/03/30 09:30 UNKN DA5COB2 WSER306
DA54000056 DEV 01.03 2005/08/09 15:33 00873 DA5COB2 BHITEFI
DA55000007 FRZ 02.02 2005/12/13 14:10 00788 DA5COB2 BHITEFI
DA55000021 FRZ 02.01 2006/01/20 03:38 00788 DA5COB2 BHITEFI
DA56000001 BAS 02.06 2006/05/22 08:16 00788 DA5COB2 BHITEF2
DA56000002 DEV 64.66 2006/11/07 03:34 00811 DA5COB2 BHITEFI
DA56000008 BAS 01.00 2006/07/26 14:01 00788 DA5COB2 BHITEFI
DA56000012 FRZ 01.02 2007/03/12 04:12 00811 DA5COB2 BHITEF2
DA57000007 BAS 02.04 2007/02/16 10:22 00808 DA5COB2 BHITEFI
DA57000001 DEV 02.01 2007/09/04 13:40 00835 DA5COB2 BHITEF2
DA57000009 BAS 02.06 2007/07/16 15:42 00835 DA5COB2 BHITEF2

```

This table describes the fields on the History Information panel.

Package ID	Displays the package ID for the package that contains the component which abended..
Status	Displays the status of the package.
Promotion	Level that the component resides after this action.
VV.MM	The version and modification number for the component following the IBM standards (mm starts at 01, increments to 99; vv starts at 01, increments to 99, and recycles to 01). The first time the component is processed by ChangeMan ZMF the vv.mm is 01.01. Each successive stage request (in the same change package) increments the mm portion (for example, 01.02, 01.03). The next package using the component causes the vv portion to be incremented (02.01).
Last Action	The date and time of the last change to the component.
Size	The size of the component.
PROCNAME	The procedure that compiled and link-edited the source file.
ID	The TSO ID of the user who performed the last action on the component.

Display ChangeMan Source Listings

The ChangeMan Source Listing panel displays the listing found in ChangeMan ZMF for the abended program.

To access the ChangeMan Source Listing panel, select option S on the main ChangeMan ZMF Information panel.

```

BROWSE      SYS04189.T114223.RA000.USER18.ESPYT.H02      Line 00000000 Col 001 080
Command ==>>>                                         Scroll ==>> PAGE
***** Top of Data *****
*****
*          DDNAME: SERCOPY.SYSRINT                      *
*****
                                IEBCOPY MESSAGES AND CONTROL STATEMENTS
IEB1135I IEBCOPY  FMID HDZ11G0  SERVICE LEVEL UA08612  DATED 20070202 DFSMS 01.0
IEB1035I RPFSTG1  SERCOPY  14:30:25 THU 01 JUL 2007 PARM='LIST=NO'
COPY  OUTDD=SYSUT2,INDD=((SYS00001,R)),LIST=NO  * Unload
SELECT MEMBER=(ESPYCOB3)
IEB1013I COPYING FROM PDS  INDD=SYS00001 VOL=SRSM7E DSN=START.SERQSTDA.Q002.STD
IEB1014I          TO PDS  OUTDD=SYSUT2  VOL=SRSM4E DSN=SYS04183.T143021.RA000.
IEB1098I 1 OF 1 MEMBERS COPIED FROM INPUT DATA SET REFERENCED BY SYS00001
IEB144I THERE ARE 14 UNUSED TRACKS IN OUTPUT DATA SET REFERENCED BY SYSUT2
IEB149I THERE ARE 0 UNUSED DIRECTORY BLOCKS IN OUTPUT DIRECTORY
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE

```

Logic for Retrieving Source Code from ChangeMan

StarTool DA uses the following logic to retrieve source code of abending programs that are controlled by ChangeMan ZMF:

- 1 StarTool DA tries to access the source in the ChangeMan ZMF staging library.
- 2 If staging library doesn't exist, StarTool DA checks if the program's package has been baselined.
- 3 If the package is baselined, StarTool DA retrieves the source code from the baseline library.

located ESxxSPRM member in the customer SHPLIB (with the ESxx value being replaced with their DA configuration identifier). See the StarTool DA Installation Guide for information on changing this parameter.

Chapter 8

Dump Examples

Here are some examples of Mini Dump Reports.

SOCB with Language Server (COBOL II)	108
S130 Abend	112
S813 (Diagnostic Section Only)	115
SOC4 - 11 (Storage Not Available)	116
SOC4 - 4 (Low Storage Access)	118
SOC4 - 4 (Storage Key Mismatch)	119
SOC4 - 4 (Storage Protected)	120
DB2 Example	121

SOCB with Language Server (COBOL II)

The following Mini Dump Debug Report is generated when a COBOL-II program abends with an SOCB. The Language Server identifies the failing machine instruction, content of error operand and source COBOL statement. (No DCBs or ACBs are printed since this program does not contain files).

```

*****
*                               StarTool DA Diagnostics                               *
*                               *                               *                       *
*                               Copyright (C) 2004, SERENA Software Inc..           *
*                               Unpublished - Rights Reserved under the             *
*                               Copyright Laws of the United States.                *
*                               *                               *                       *
*****

*                               Job Information                               *
*                               *                               *                       *
Completion Code : S0CB-0B
Jobname         : WST0069
Stepname        : ESPYE
Job Owner       : WST006
Time            : 06:58:22

*                               System Information                               *
*                               *                               *                       *
CPU MODEL       : 7490
Operating System : OS/390 6.0.5
Control Program FMID : HBB6605
SMF System ID    : P390
DFSMS            : V1.4.0
JES2             : OS 2.5.0

*                               Diagnostic Section                               *
*                               *                               *                       *
SYSTEM ABEND: SOCB

Description:  This ABEND is a decimal divide exception. A quotient
              exceeds the specified data field size.

User Action:  Dividing by zero is the most common cause of this ABEND.
              Correct the program logic error that caused the divide
              exception and rerun the job.

*                               Failing Instruction Section                               *
*                               *                               *                       *

The Failing Instruction is in Program (CSECT):
DOIT9   At Offset +X'0002DE'

The failing instruction is:
====> DP   X'160'(6,R13),X'28'(3,R9)   <====

*                               Failing Instruction Analysis                               *
*                               *                               *                       *
*                               Instr Type = SS   Number of Operands = 2           *
* Operand 1 - Type = DATA  ADDR = 00008CB0 DATA = 00000049 994C               *
DIVIDE BY ZERO--> * Operand 2 - Type = DATA  ADDR = 00008CE0 DATA = 00000C           *
*****

*                               Failing COBOL Statement Section                               *
*                               *                               *                       *

**** The failing Instruction is within COBOL Statement:           **

====>000035 PROCEDURE DIVISION.                                     <==
**** Operand 1 ('TO' field) definition is as follows:
====> *** Field is compiler defined. No Data Division definition exists.      <==
      Operand 1 value = 00000049 994C

**** Operand 2 ('FROM' field) definition is as follows:
====> 000022 77 BAD-DATA PIC S9(5) COMP-3 VALUE 00000.                 <==
      Operand 2 value = 00000C

*****
*                               Instruction sequence prior to and including abending instruction.                               *
*                               *                               *                       *

```

```

*
*   Address      Offs      Mach Code      Opcode      Operand      Data
*   -----
*   00008AD2    002A2    8E40 0020      SRDA      R4,X'20'
*   00008AD6    002A6    4820 A006      LH        R2,6(,R10)
*   00008ADA    002AA    8E20 0020      SRDA      R2,X'20'
*   00008ADE    002AE    1A42          AR        R4,R2
*   00008AE0    002B0    1E53          ALR       R5,R3
*   00008AE2    002B2    47C0 B15A      BNP       X'15A'(,R11)
*   00008AE6    002B6    5A40 C004      A        R4,4(,R12)
*   00008AEA    002BA    5D40 C000      D        R4,0(,R12)
*   00008AEE    002BE    5040 9078      ST       R4,X'78'(,R9)
*   00008AF2    002C2    5830 9090      L        R3,X'90'(,R9)
*   00008AF6    002C6    5B30 9078      S        R3,X'78'(,R9)
*   00008AFA    002CA    1823          LR        R2,R3
*   00008AFC    002CC    8E20 0020      SRDA      R2,X'20'
*   00008B00    002D0    5D20 C000      D        R2,0(,R12)
*   00008B04    002D4    5020 9080      ST       R2,X'80'(,R9)
*   00008B08    002D8    F852 D160 9030  ZAP      X'160'(6,R13),X'30'(3,R9)
*   00008B0E    002DE    F52D D160 9028  DP       X'160'(6,R13),X'28'(3,R9)
*
*****

```

The program was compiled on 21 Jan 1999 By PGMID 5668-958-01, V 1.4 M 00 (COBOL II)
It is a CSECT in Load Module DOIT9 Compile Time Was: 10:35:28
The module was loaded from STEPLIB Library ULTIMIS.ULTITEST.ISPLLIB
It was Link Edited on 21 Jan 1999

* PSW and Register Section *

ABEND PSW - 078D2000 00008B14 A(DOIT9) + 0002E4
Prog PSW - 078D2000 00008B14 A(DOIT9) + 0002E4
Load Module - DOIT9 Entry Point Address - 00008830
ILC - 06, INTC - 0B Load Point Address - 00008830

Registers at time of error (Descriptions based on resolutions as 24 bit addresses).

Register : 00
Hex Value : 00000100
Decimal Value : 256
Points to : Load Point of PSA + 000100

Register : 01
Hex Value : 00016CF4
Decimal Value : 93,428
Points to : Address 00016CF4 + 000CF4 (Private Area, Subpool 000, ALLO)

Register : 02
Hex Value : FFFFFFFE3
Decimal Value : - 29
Points to : Load Point of PVT + 032FE3

Register : 03
Hex Value : 00000000
Decimal Value : 0
Points to : Load Point of PSA + 000000

Register : 04
Hex Value : 0000001D
Decimal Value : 29
Points to : Load Point of PSA + 00001D

Register : 05
Hex Value : 00000000
Decimal Value : 0
Points to : Load Point of PSA + 000000

Register : 06
Hex Value : 00008884
Decimal Value : 34,948
Points to : Load Point of DOIT9 + 000054

Register : 07
Hex Value : 00010260
Decimal Value : 66,144
Points to : Load Point of DOIT9 + 007A30

Register : 08
Hex Value : 0000F06C
Decimal Value : 61,548
Points to : Load Point of DOIT9 + 00683C

Register : 09
Hex Value : 00008CB8
Decimal Value : 36,024
Points to : Load Point of DOIT9 + 000488

Register : 10
Hex Value : 000088D4
Decimal Value : 35,028
Points to : Load Point of DOIT9 + 0000A4

Register : 11

Chapter 8: Dump Examples

```

Hex Value      : 00008990
Decimal Value  : 35,216
Points to     : Load Point of DOIT9 + 000160

Register      : 12
Hex Value     : 000088B0
Decimal Value : 34,992
Points to    : Load Point of DOIT9 + 000080

Register      : 13
Hex Value     : 00008B50
Decimal Value : 35,664
Points to    : Load Point of DOIT9 + 000320

Register      : 14
Hex Value     : 00008A14
Decimal Value : 35,348
Points to    : Load Point of DOIT9 + 0001E4

Register      : 15
Hex Value     : 00000000
Decimal Value : 0
Points to    : Load Point of PSA + 000000
    
```

* Call Trace Summary *

```

**** Caller ****          ***** At *****          ***** Calls *****
Load-Mod Program          Offset          Program Load-Mod
*SYSTEM*                  +              DOIT9 DOIT9
DOIT9                      + @0IDE          -- Calls IGENRI DOIT9
                        -- Calls
    
```

```

*****
* Application Program Attributes *
Load-Mod Program Compile Date Length Language
DOIT9 DOIT9 21 Jan 1999 000551 COBOL II
DOIT9 IGENRI 13 May 1993 000D88 COBOL II
    
```

```

* ----- Virtual Storage Print for Program DOIT9 ----- *
| ADDR | DISP |----- D A T A -----| |----- TRANSLATION -----|
00008830 00000 47F0F070 23C4D6C9 E3F94040 4040C3F2 40F14BF4 4BF040F0 F161F2F1 61F9F940 00DOIT9C2.1.4.0 01/21/99
00008850 00020 F1F04BF3 F54BF2F8 00008884 E0E80C4C 00000000 00840108 00000000 08000000 10.35.28hdY<daa
00008870 00040 00000000 0000000B 00000000 00000000 40404040 00008830 000088B0 00008B50 hUh^ &
00008890 00060 00008835 00008830 00008A14 0000AB38 90EC000C 5810F028 98EFF068 07FF0000 hhø™_0iQu0 -É
000088B0 00080 05F5E100 00000001 00000000 40404040 40404040 40404000 000C4040 000088D4 5 fm

* ----- Virtual Storage Print for Program DOIT9 ----- *
| ADDR | DISP |----- D A T A -----| |----- TRANSLATION -----|
000088D0 000A0 00008990 00008A14 00000001 00000000 00000008 00000014 00000048 00001388 iãïh
000088F0 000C0 0047E2E3 C1D9E340 D6C640D9 C5C3C1C4 C4D97E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E STARTOFRECADDR=====
00008910 000E0 7E7EC4D6 C9E3F240 E6D6D9D2 C9D5C740 E2E3D6D9 C1C7C5C4 D6C9E3F9 404040E2 ==DOIT2WORKINGSTORAGEIDOIT9 S
00008930 00100 E8E2D6E4 E34040D6 D540D6C6 C649994C 00000000 0000013C 00000001 00000140 Y$OUTONOFFr<°
00008950 00120 00000001 00000000 00000000 80000000 40800000 000025C0 00014000 08080000 Å { åa
00008970 00140 1C005380 00000000 40C00001 40000508 00001C00 53024000 08080000 1C005300 % ååå
00008990 00160 D2149000 A03ED20A 9015C00C D2029020 A066D202 9028C017 D2029030 A069D21F KkKæMkøi{ÈKøÙæK
000089B0 00180 9038A01E D2039058 A018D203 9060A014 D2039068 A010D203 9070A00C D2039078 KK-Kø æKø°æK øÅ
000089D0 001A0 C008D203 9080C008 D2039088 C008D203 9090C008 5820D138 D2032000 A008D203 KkKøø{å ·JåK·æåK
000089F0 001C0 2008A008 D2032010 A008D207 2018A05B D2032020 A008920E 2028920F 20305830 KkSk··æåK·Ï·U·Ü
00008A10 001E0 D14407F3 58A0C01C 5890D13C D203D0EC A0049180 D14C58B0 C0204780 B0AC5820 J3JkæjJåU<^ {·FÅ^0·
00008A30 00200 D0C5C8F0 21CC4110 A09F05EF 9680D14C 9140D14C 4710B0D0 D203D148 D1444120 *0øJ<j J<ç^}KJçJ·y·
00008A50 00220 B0CA5020 D14447F0 B000D203 D144D148 9640D14C D5029020 A06358B0 C0204770 &JØKJJo J<Nø·æ ^{·†°
00008A70 00240 B1385820 90605A20 90581842 8E400020 4820A016 8E200020 1B421F53 4730B104 -!ç·æE·øfç†Uæ†
00008A90 00260 5B40C004 5D40C000 50409088 D2029020 A0664830 A01C5A30 90881823 8E200020 $) &hKæMkøi{Uæ!Uøh»È·
00008AB0 00280 5D20C000 50209090 D2039058 9090D203 9060C008 47F0B178 58209070 5A209068 )&Kkø·{å†øEÅ·ø°!·ø_
00008AD0 002A0 18428E40 00204820 A0068E20 00201A42 1E5347C0 B15A5A40 C0045D40 C0005040 -ç†{æ!! {i} {&
00008AF0 002C0 90785830 90905B30 90781823 8E200020 5D20C000 50209080 F852D160 9030FD52 $)·{&·øÅ8^J·øÙø^
00008B10 002E0 D1609028 F8229030 D160D201 D07AC004 47F0B1AA 9120D054 47E0B1AA 5820D05C J-8J-K:†øæ-j·}ç†æ-·.}*
00008B30 00300 58F021CC 4110A08D 05EF947F D14C5820 D0C5C8F0 22204110 A08C05EF 00000000 0m° J<ç* ø°·yæE ü
00008B50 00320 00108001 00005FA0 00005BA0 00008A14 0000AB38 80006A84 00016CF4 00000040 °$~åÅ°d%4
00008B70 00340 008CB09C 008CB078 008DCA70 008BDF58 FD000000 008DC9D8 008DC688 00000000 8øçIQçFh00008B90 00360 008DCA70 00E965EA
00360 C3F2E3C7 E34EF4F8 03000000 61100020 00016CC0 00010260 ZC2TGT+48/·%{-
00008B90 00380 00000000 00000000 00000094 00000000 00000000 00000000 00000000 00000000 m
00008BD0 003A0 00000000 00000000 30300000 00000000 00000001 00000000 00000000 E2E8E2D6 E4E34040

SYSOUT
00008BF0 003C0 C9C7E9E2 D9E3C3C4 00000000 00000000 00000000 00000000 00000000 00000000 IGSZRTCD
00008C10 003E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00008C30 00400 00000000 00000000 00000000 00000001 00008C9C 00000000 00008940 00000000 Æ°i
00008C50 00420 00008830 000088D4 00008C94 000088D0 00000000 00008CB8 00000000 40404040 hhMmhæK
00008C70 00440 40404040 40404040 40404040 40404040 40404040 40404040 00008D50 00008CB8 Ç&EæK
00008C90 00460 00000000 00008A14 00008A14 C0000000 00000000 00000000 00000000 00000000 00000000
00008CB0 00480 00000049 994C0000 C4D6C9E3 F240E6D6 D9D2C9D5 C740E2E3 D6D9C1C7 C5404040 r<DOIT2WORKING STORAGE
00008CD0 004A0 40404040 40404040 D6C6C600 00000000 00000C00 00000000 49994C00 00000000 OFFçr<
00008CF0 004C0 E2E3C1D9 E340D6C6 40D9C5C3 C1C4C4D9 7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E STARTOFRECADDR=====
00008D10 004E0 00001388 00000000 00000048 00000000 00000014 00000000 00000008 00000000 hiã
00008D30 00500 0000001D 00000000 FFFFFFE3 00000000 00000000 00000000 00000000 00000000 T
00008D50 00520 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00008D70 00540 00000000 00000000 00000000 00000000 0F000000
* ----- Virtual Storage Print for Program IGENRI ----- *
| ADDR | DISP |----- D A T A -----| |----- TRANSLATION -----|
0000AB38 00000 47F0F034 2740C9C7 E9C5D5D9 C9404200 C3F2F44B F0F0F561 F1F361F9 F340F1F7 00IGZENRIC24.005/13/93 17
0000AB58 00020 4BF1F340 4040D600 0000AB64 E4D5F4F2 F5F8F840 18BD58D0 100890EC D00C18CF .130UN42588 b·jåø™}i
0000AB78 00040 5830CC60 12334770 C04E5830 CC649180 303A47E0 C07A58F0 CC6812FF 4780C07A --+jüè†{·ø° E†Å{·
0000AB98 00060 D501D000 CC8C4770 C0769101 D0034700 00769604 D05618DB 07FF50B0 D00450D0 Nj{Ao!}E-È&^}å&
0000ABB8 00080 B008D501 D000CC8C 4770C0A4 9101D0E3 47E0C0A4 5880B018 5080D018 5880B040 Nuç†\u·Å^&Å·Å^
0000ABD8 000A0 5080D040 58B0CC28 41700114 1EB75880 CC344130 002C1E83 41300010 18615830 &°pÿUN-çyÜ/_U
    
```

```

0000ABF8 000C0 30004170 B0B49180 3074189D 4780C0E4 9610B058 96018012 91403074 4780C0E4 jUo^ oAj U%tA{U
0000AC18 000E0 9604B059 58D0D004 1F33D501 D000CC8C 4770C2AE 9101D003 47E0C2AE 9180D002 oNt°B0j}t\B0jA}
0000AC38 00100 47E0C118 5840D100 9120402E 47E0C114 41300010 47F0C1FA 5840D004 D5014000 AJjAyU0tAU }iN
0000AC58 00120 CC8C4770 C1D29101 400347E0 C1D29180 400247E0 C1D25850 41009120 502E47E0 AKjAKj t\AK &yj ·&t\
0000AC78 00140 C14A4130 001047F0 C1FA90EC D00C41E0 C1C84100 00681110 0A0A05F0 50D10004 A>0AAHy 00 0&Ji
0000AC98 00160 501D0008 18D14120 D048D21F 2000F054 50E2000C 181295E8 E0044770 F03A0700 &JK0&snY\it°0é-
0000ACB8 00180 45F0F038 0000ACC4 00000000 C9C7E9F9 E6E3D640 0A060700 45F0F050 0000ACDC 00DIGZ9WTO 0É-00&0_
0000ACD8 001A0 00000000 C9C7E9C5 E6E3D640 0A060000 00000063 00000002 00000004 00000000 IGZEWTO i
0000ACF8 001C0 00000007 0000AB3E F1F0F140 D50047F0 C1FA9110 D00247E0 C1E24130 000847F0 101N0AUj}t\ASyUat0
*----- Virtual Storage Print for Program IZGENRI -----*
| ADDR | DISP |----- D A T A -----| |----- TRANSLATION -----|
0000AD18 001E0 C1FA90EC D00C41E0 C1F458F0 C1F007FF 0000AC8A F1F0F240 D5004150 00101535 AA40A00@102 Ny&As
0000AD38 00200 4780C20E 41500008 15354770 C2AE9110 D0024780 C21E5880 D03447F0 C2629180 B&Bj}tAB- A}p0B2jA
0000AD58 00220 D0024780 C2325840 D0585880 404047F0 C2625840 D0049180 40024780 C24A5850 B0B' }ijA tAB, &
0000AD78 00240 40585880 504047F0 C26290EC D00C41E0 C25C58F0 C25807FF 0000AC8A F1F0F340 &0BB* 0B -É0@103
0000AD98 00260 D50058D0 808CD703 D000D000 9610D001 9601D003 9610D002 5870801C D707D058 NPoo}0} °AP-}
0000ADB8 00280 D0584150 00101535 4770C298 41400005 4040D05C 47F0C2A0 41500004 4050D05C &Bq }*t0B&y&i &}*
0000ADD8 002A0 9618D058 4110D058 58F071CC 05EF58D0 D0089140 90544780 C2CC5830 60041233 o0}aj 0StAB* U-iY
0000ADF8 002C0 4780C2CC 91609054 47E0C5F8 58308020 12334770 C3D09102 800C4770 C3D0D503 Bj -EYt°C}jAt°C}N
0000AE18 002E0 6000CC20 4780C3D0 5810CC60 12114770 C2F65810 CC649104 103847E0 C3D05830 -C-B6 "jiAtC}U
0000AE38 00300 D0045830 30041233 4780C3D0 1F335840 CC6C1244 4770C31E 41E00001 1E3E58F0 C"%t°C-y\ -0
0000AE58 00320 CC7012FF 4770C32E 41500002 1E351233 4780C358 D203D060 CC88BFEE D06016E3 C&tAC K}-"hçü)-T
0000AE78 00340 BEEFD060 4130D060 90ECD00C 18E358F0 C35407FF 0000AC8A 58306004 12334780 --T0CS-É0 U-iYtA
0000AE98 00360 C3CC4100 00484510 C36A0A0A 1F335030 10005010 D00850D0 10045030 100818D1 CC*&&i&i&Uaj
0000AEB8 00380 41100001 41F0C866 05EF4100 0048181D 58D0D004 41101000 0A0A5830 90045830 0H }iY00 U0i U
0000AED8 003A0 30041233 58A08020 4770C3B4 5090A068 47F0C3BC 58309004 5030A068 5830A068 C&t0Cä U0i&Uae Uae
0000AEF8 003C0 58303004 5030A018 47F0C3D0 9602800C 58308020 12334770 C5EA4130 B0B45030 &0Co UA·Yt°EyU^V&U
0000AF18 003E0 B05C41D0 8034D703 D048D048 50B0D048 41100048 1E1D58F0 71EC05EF D2078000 *P&y±- 0 °m üK-Ä
0000AF38 00400 CC784130 01F85030 800850B0 80185070 801C58B0 CC384130 002C1EB3 5830B00C 8&&&A °"äyÜN-ú U^
0000AF58 00420 50308198 5830B010 5030819C 5830B014 503081A0 5830B024 50308180 9688800C &aq&a&Uae U^A&UaAohA
0000AF78 00440 5810CC64 D2008124 1026D202 81251038 D2018128 104645E0 CA304120 040058F0 KaKaKaI0\èÜy·i 0
0000AF98 00460 801C0700 1BEE0700 4510C474 28800000 00000002 41110000 58F0F1F4 05EF5010 Dy 014 ü&
0000AFB8 00480 808C4120 01540700 1BEE41E0 81AC58FE 000012FF 4780C50E 18124111 003F5410 aEtAEy

```

***** E N D O F S T A R T O O L D A D U M P D A T A *****

S130 Abend

The following Mini Dump Debug Report (only a portion is shown) is generated when an S130 type abend is encountered.

```
*****
*      Job Information      *
*****
```

```
Completion Code : S130
Jobname         : WST006BR
Stepname        : ESPYE
Job Owner       : WST006
Time            : 11:04:49
```

```
*****
*      System Information  *
*****
```

```
CPU MODEL       : 7490
Operating System : OS/390 6.0.5
Control Program FMID : HBB6605
SMF System ID    : P390
DFSMS           : V1.4.0
JES2            : OS 2.5.0
```

```
*****
*      Diagnostic Section  *
*****
```

SYSTEM ABEND: S130

Description: A DEQ was issued for a resource name that had not previously been ENQ'ed by the task that issued the DEQ. This is probably due to a logic error in the program that issued the DEQ.

User Action: Make sure your program owns a resource before DEQ'ing it, or use the RET=HAVE keyword on the DEQ macro. If you are analyzing a dump of this abend, the UNIQUE-3 column in the trace table entry for the SVC 30 or SVC 38 will contain the address of the DEQ parm list. Check all addresses in the parm list for correct format and usage.

```
*****
*      Failing Instruction Section      *
*****
```

The Failing Instruction is in Program (CSECT):

ISGGABND At Offset +X'00130A'

The failing instruction is:

```
====> SVC 13 <====
```

```
*****
*      This is a code S130 Abend. It is the result of an error in issuing a(n) DEQ SVC.
*      The SVC that caused the abend is in LMOD TIMEFMT6, CSECT TIMEFMT6, at offset X'000B20.
*      The following code reflects the sequence of instructions leading up to the failing SVC Call.
*****
```

```
*****
*      Instruction sequence prior to and including the failing SVC Call.
*****
```

Address	Offs	Mach Code	Opcde	Operand	Data
00007AD0	00AE8	960F C04F	OI	X'4F'(R12),X'0F'	
00007AD4	00AEC	F377 ACC3 C048	UNPK	X'CC3'(8,R10),X'48'(8,R12)	
00007ADA	00AF2	DC07 ACC3 AC2B	TR	X'CC3'(8,R10),X'C2B'(R10)	
00007AE0	00AF8	4110 AB78	LA	R1,X'B78'(,R10)	
00007AE4	00AFC	4100 AC77	LA	R0,X'C77'(,R10)	
00007AE8	00B00	1FFF	SLR	R15,R15	
00007AEA	00B02	BFF7 1031	ICM	R15,7,X'31'(R1)	
00007AEE	00B06	05EF	BALR	R14,R15	
00007AF0	00B08	4160 6010	LA	R6,X'10'(,R6)	
00007AF4	00B0C	47F0 AA82	B	X'A82'(,R10)	
00007AF8	00B10	4510 AB20	BAL	R1,X'B20'(,R10)	
00007AFC	00B14	DC	DC	X'C02C480000010230'	{N+U
00007B04	00B1C	DC	DC	X'00007BC8'	#H
00007B08	00B20	0A30	SVC	48	

TIMEFMT6 was compiled on 15 Sep 2002 By PGMID 5696-234-00, V 0.1 M 02 (HLA ASMBL)
It is a CSECT in Load Module TIMEFMT6
The module was loaded from STEPLIB Library WST006.LOADLIB

It was Link Edited on 15 Sep 2002

```

*****
* PSW and Register Section *
*****
ABEND PSW - 070C1000 814818B2 A(ISGGNQDQ) + 001C0A
Prog PSW - 070C0000 00007B0A A(TIMEFMT6) + 000B22
Load Module - ISGGNQDQ Entry Point Address - 0147FCA8
ILC - 02, INTC - 0D Load Point Address - 0147FCA8
*****
* This is an Abend Code S130. For this type of abend, the registers at time *
* of error are virtually useless. The following are the regs at the time *
* the DEQ Macro was issued. *
*****
Registers at time of error (Descriptions based on resolutions as 24 bit addresses).

Register : 00
Hex Value : 00007C5F
Decimal Value : 31,839
Points to : Load Point of TIMEFMT6 + 000C77

Register : 01
Hex Value : 80007AFC
Decimal Value : -2,147,452,164
Points to : Load Point of TIMEFMT6 + 000B14

Register : 02
Hex Value : FFFFFFF9
Decimal Value : -7
Points to : Load Point of PVT + 032FF9

Register : 03
Hex Value : 00007C35
Decimal Value : 31,797
Points to : Load Point of TIMEFMT6 + 000C4D

Register : 04
Hex Value : 008CBD78
Decimal Value : 9,223,544
Points to : Load Point of PVT + 8CBD78

Register : 05
Hex Value : 008DCA70
Decimal Value : 9,292,400
Points to : Address 008DC000 + 000A70 (Local SQA , Subpool 255, ALLO)

Register : 06
Hex Value : 00008190
Decimal Value : 33,168
Points to : Load Point of TIMEFMT6 + 0011A8

Register : 07
Hex Value : FD000000
Decimal Value : -50,331,648
Points to : Load Point of PSA + 000000

Register : 08
Hex Value : 008DC9D8
Decimal Value : 9,292,248
Points to : Address 008DC000 + 0009D8 (Local SQA , Subpool 255, ALLO)

Register : 09
Hex Value : 00007D48
Decimal Value : 32,072
Points to : Load Point of TIMEFMT6 + 000D60

Register : 10
Hex Value : 80006FE8
Decimal Value : -2,147,455,000
Points to : Load Point of TIMEFMT6 + 000000

Register : 11
Hex Value : 008DCA70
Decimal Value : 9,292,400
Points to : Address 008DC000 + 000A70 (Local SQA , Subpool 255, ALLO)

Register : 12
Hex Value : 00007D48
Decimal Value : 32,072
Points to : Load Point of TIMEFMT6 + 000D60

Register : 13
Hex Value : 00009800
Decimal Value : 38,912
Points to : Address 00009000 + 000800 (Private Area, Subpool 069, ALLO)

Register : 14
Hex Value : 50007AF0
Decimal Value : 1,342,208,752
Points to : Load Point of TIMEFMT6 + 000B08

Register : 15
Hex Value : 00D1E6F0

```

Decimal Value : 13,756,144
Points to : Load Point of IGG019DJ + 000018

* Call Trace Summary *

**** Caller ****	***** At *****	**** Calls ****
Load-Mod Program	Offset	Program Load-Mod
SYSTEM	+	-- Calls TIMEFMT6 TIMEFMT6
TIMEFMT6 TIMEFMT6	+ 0B06	-- Calls IGG019DJ IGG019DJ

* Application Program Attributes *

Load-Mod	Program	Compile Date	Length	Language
TIMEFMT6	TIMEFMT6	15 Sep 2002	002018	HLA ASMBL
IGG019DJ	IGG019DJ	23 Apr 1997	000928	ASSMBLR H

***** E N D O F S T A R T O O L D A D U M P D A T A *****

S813 (Diagnostic Section Only)

In this example, the file name in the JCL did not match the name on the tape label. See the Action item at the end of this example for the resolution to this problem.

Completion Code - S813 - 04

```
*****
* Diagnostic Section *
*****
```

SYSTEM ABEND: S813 -04

Description: The error happened when an OPEN macro was issued for a data set on tape. This abend is accompanied by message IEC149I, which explains the return code issued with the ABEND code.

User Action: This ABEND usually indicates a mismatch between the data set name specified in the JCL and the data set name in the tape label for the data set. Refer to message IEC149I for a complete description of the ABEND return code.

*** Additional information for this item appears below: ***

Message Format: IEC149I 813-rc,mod,jjj,sss,ddn{-#},ser,DSN

Example: IEC149I 813-04,IFG0032,Jobname,Stepn,DDname,CART01,DSName

Description: An error occurred during OPEN processing for a data set on a magnetic tape. The 'rc' value in the message is a return code that describes the error in detail. Possible 'rc' values and their meanings are provided below.

For rc=04, an OPEN was issued for a data set on magnetic tape, but the data set name in the tape's header label did not match the data set name in the JCL. If the header label does match the JCL data set name, verify that the JFCB was not modified incorrectly. This message can also occur if the first file of a magnetic tape does not exist (that is, the tape has a volume label and a dummy header).

User Action: Ensure that the data set name and volume serial number on the DD statement are correct; if they are, verify that the JFCB was not modified incorrectly by the program that issued the OPEN.

```
*****
* Additional ABEND Information *
*****
```

Abending DDNAME is TAPE

DSNAME from DD card and DSNAME on Tape
(VOLSER=HDS260) do not agree.
DSNAMES from DD card and Tape Label:

DD CARD: SP1SYS.DUMP01
TAPE : TWORK.Y2K.INSTALL

```
*****
* NOTE: Only the last 17 bytes of the *
* TAPE label and the DD DSN are compared.*
*****
```

Action: Change the DD CARD DSN to match the TAPE DSN - or
Specify the TAPE VOLSER whose DSN matches the DD CARD DSN.

SOC4 - 11 (Storage Not Available)

In this abbreviated example, an assembler program executing in a subspace referred to storage that was not available to it. A program in a subspace can only refer to storage in the subspace or storage common to all subspaces (reason code 10 or 11). The assembler failing operand is identified and tells you the reason the operand failed. StarTool DA provided critical information.

Completion Code - S0C4 - 11

* Diagnostic Section *

SYSTEM ABEND: S0C4

Description: This ABEND is caused by a hardware detected virtual address translation error, or a storage protection violation. One of the following is true:

- The requested referenced was never allocated (i.e., never GETMAIN'ed - reason code 10 or 11)
- this is a page-translation or segment-translation exception.

- The requested virtual storage was paged out and the routine that referenced it was disabled for I/O interrupts (reason code 11 with disabled PSW).

- A program executing in a subspace referred to storage that was not available to it. A program in a subspace can only refer to storage in the subspace or storage common to all subspaces (reason code 10 or 11).

User Action: Correct the program logic error that generated the invalid address or storage reference. When analyzing the dump, remember that the PSW saved when an 0C4 abend occurs may point at the failing instruction or it may point at the next instruction after the failing instruction.

Check to ensure that your program is obtaining, using, and freeing storage properly. Moving data to a zero address or to an address less than 512 (decimal) is a very frequent cause of this abend.

Possible causes for this abend include:

COBOL: Invalid address was referenced due to subscript error or bad parameter passed
In group move, receiving record variable length defined incorrectly
Tried moving variable length record that was larger than target field size
Tried to read or write a file which was not open
Used DD DUMMY with logic that moves high values to FD
Tried to call within COBOL SORT I/O procedure
Tried to "goback" in the SORT output procedure

 * Failing Instruction Section *

The Failing Instruction is in Program (CSECT):

TIMEFMT6 At Offset +X'0001E6'

The failing instruction is:

==> MVC 0(8,R6),X'B40'(R10) <==

 * Failing Instruction Analysis *

 * Instr Type = SS Number of Operands = 2 *
 * Operand 1 - Type = DATA ADDR = 00009000 DATA = *
 STG NOT AVAIL ---> * Operand 2 - Type = DATA ADDR = 00006BB8 DATA = FFFFFFFF FFFFFFFF *

 * Instruction sequence prior to and including abending instruction. *

Address	Offs	Mach	Code	Opcde	Operand	Data
00006222	001AA	4100	ACD0	LA	R0,X'CD0'(:,R10)	
00006226	001AE	5000	1004	ST	R0,4(:,R1)	
0000622A	001B2	4100	C20C	LA	R0,X'20C'(:,R12)	
0000622E	001B6	5000	1000	ST	R0,0(:,R1)	
00006232	001BA	41D0	D000	LA	R13,0(:,R13)	
00006236	001BE	58F0	0010	L	R15,X'10'	
0000623A	001C2	58F0	F128	L	R15,X'128'(:,R15)	
0000623E	001C6	58F0	F064	L	R15,X'64'(:,R15)	
00006242	001CA	58F0	F030	L	R15,X'30'(:,R15)	
00006246	001CE	0CEF		BASSM	R14,R15	
00006248	001D0	5870	5000	L	R7,0(:,R5)	
0000624C	001D4	1817		LR	R1,R7	
0000624E	001D6	4160	C2B4	LA	R6,X'2B4'(:,R12)	
00006252	001DA	5A60	ACC8	A	R6,X'CC8'(:,R10)	
00006256	001DE	B100	6000	LRA	R0,0(:,R6)	
0000625A	001E2	4780	A1DA	BZ	X'1DA'(:,R10)	
0000625E	001E6	D207	6000 AB40	MVC	0(8,R6),X'B40'(R10)	

SOC4 - 4 (Storage Key Mismatch)

In this example, the protect key in the assembler program's PSW does not match the protect key of a storage area. The assembler failing operand is identified and tells you the exact reason for the failed operand.

Completion Code - S0C4 - 04

* Failing Instruction Section *

The Failing Instruction is in Program (CSECT):

TIMEFMT6 At Offset +X'0001DA'

The failing instruction is:

==> ST R6,0(,R1) <==

* Failing Instruction Analysis *

* Instr Type = RX Number of Operands = 2 *
* Operand 1 - Type = REG VALU = 00007000 REG# = 06 *
STG KEY MISMTCH-->* Operand 2 - Type = ADDR ADDR = 0563C008 = *

* Instruction sequence prior to and including abending instruction. *

Address	Offs	Mach	Code	Opcde	Operand	Data
00006206	0019E	4100	C20C	LA	R0,X'20C'(,R12)	
0000620A	001A2	5000	1000	ST	R0,0(,R1)	
0000620E	001A6	41D0	D000	LA	R13,0(,R13)	
00006212	001AA	58F0	0010	L	R15,X'10'	
00006216	001AE	58F0	F128	L	R15,X'128'(,R15)	
0000621A	001B2	58F0	F064	L	R15,X'64'(,R15)	
0000621E	001B6	58F0	F030	L	R15,X'30'(,R15)	
00006222	001BA	0CEF		BASSM	R14,R15	
00006224	001BC	5870	5000	L	R7,0(,R5)	
00006228	001C0		1817	LR	R1,R7	
0000622A	001C2	4160	C2B8	LA	R6,X'2B8'(,R12)	
0000622E	001C6		0700	NOPR	R0	
00006230	001C8	4510	A1D0	BAL	R1,X'1D0'(,R10)	
00006234	001CC			DC	X'00000024'	A
00006238	001D0	5810	1000	L	R1,0(,R1)	
0000623C	001D4		0A6B	SVC	107	
0000623E	001D6	5810	ACD0	L	R1,X'CD0'(,R10)	
00006242	001DA	5060	1000	ST	R6,0(,R1)	

The program was compiled on 22 Jan 1999 By PGMID 5696-234-00, V 01 M 02 (HLA ASMBL)
It is a CSECT in Load Module TIMEFMT6
The module was loaded from STEPLIB Library WST006.LOADLIB
It was Link Edited on 22 Jan 1999

DB2 Example

Completion Code - U05C1

```
*****
* Diagnostic Section *
*****
```

Abend Code U05C1 not found in Abend Code File.

```
*****
* Failing Instruction Section *
*****
```

The Failing Instruction is in Program (CSECT):

CEEHABD At Offset +X'00012C'

The failing instruction is:

```
===> SVC 13 <===
*****
* Failing Instruction Analysis *
*****
*
* Instr Type = RX Number of Operands = 2
* Operand 1 - Type = REG VALU = 00000000 REG# = 15
* Operand 2 - Type = ADDR ADDR = 00000000 DATA = 00000000
*****
```

```
*****
* DB2 Data Section *
*****
```

** ----- SQL Call Summary -----

```
**
** DB2 Subsystem Name(DSN1) Plan Name(DSN8BH5C)
** Program Name(DSN8BC3C) Statement Number(439)
** SQL Timestamp(16529CBE059066A6) Address of CAB(0004A930)
** Address of SQL Call(8004758E) Address of SQL ParmList(0700CA38)
**
** DSNT408I SQLCODE = -818, ERROR: THE PRECOMPILER-GENERATED TIMESTAMP
** 16529CBE059066A6 IN THE LOAD MODULE IS DIFFERENT FROM THE BIND
** TIMESTAMP 165299470E26A76E BUILT FROM THE DBRM DSN8BC3C
** DSNT418I SQLSTATE = 51003 SQLSTATE RETURN CODE
** DSNT415I SQLERRP = DSNXEPM SQL PROCEDURE DETECTING ERROR
** DSNT416I SQLERRD = -200 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
** DSNT416I SQLERRD = X'FFFFFFFF38' X'00000000' X'00000000'
** X'FFFFFFFF' X'00000000' X'00000000' SQL DIAGNOSTIC
** INFORMATION
```

** -----Plan Information-----

```
**
** Plan(DSN8BH5C) Creator(WST004)
** Bound by(WST004) Qualifier(DSN8510)
** Bind Date(1999-07-22) Bind Time(08.05.47.621942)
** Validate(Run Time) Isolation(Cursor Stability)
** Valid(Yes) Operative(Yes)
** Acquire(Use) Release(Commit)
** Current Data(Yes) Defer Prepare(No)
** Current Server() Degree(1)
** SQL Rules(DB2) Disconnect(Explicit)
** Group Member() Dynamic Rules(Static)
** Optimize Variable Input Values(No) Keep Dynamic(NO)
```

```
Supporting Environmental Data
ABEND PSW - 078D1000 86F50EB4 A(CEEPLPKA) + 03A7AC
Prog PSW - 078D1000 86FDB778 A(ESPY9TRM) + 0001A8
Load Module - CEEPLPKA Entry Point Address - 86F16708
ILC - 00, INTC - 0D Load Point Address - 06F16708
```

Registers at time of error (Descriptions based on resolutions as 31 bit addresses).

REG	HEX	Decimal	Description
R00	84000000	-2,080,374,784	A(IOSCONSL) + 004000
R01	840005C1	-2,080,373,311	A(IOSCONSL) + 0045C1
R02	0002591C	153,884	A(0002591C) + 00491C, PVT SP=001 ALLO
R03	07008F74	117,477,236	A(07008F74) + 000F74, PVT SP=001 ALLO
R04	00025538	152,888	A(00025538) + 004538, PVT SP=001 ALLO
R05	000148B0	84,144	A(000148B0) + 0018B0, PVT SP=001 ALLO
R06	00045738	284,472	A(00045738) + 024738, PVT SP=001 ALLO
R07	00000000	0	A(PSA) + 000000
R08	80047590	-2,147,191,408	A(IGZCFCC) + 000298
R09	07008D70	117,476,720	A(07008D70) + 000D70, PVT SP=001 ALLO
R10	00025918	153,880	A(00025918) + 004918, PVT SP=001 ALLO
R11	86F50D88	-2,030,760,568	A(CEEPLPKA) + 03A680
R12	00015898	88,216	A(00015898) + 002898, PVT SP=001 ALLO
R13	00025920	153,888	A(00025920) + 004920, PVT SP=001 ALLO

R14 8001E892 -2,147,358,574 A(8001E892) + 004892, PVT SP=001 ALLO
R15 00000000 0 A(PSA) + 000000

* Call Trace Summary *

**** Caller ****	**** At ****	**** Calls ****
Load-Mod Program	Offset	Program Load-Mod
SYSTEM	+	-- Calls TESTCB3C TESTCB3C

* Application Program Attributes *
Load-Mod Program Compile Date Length Language
TESTCB3C TESTCB3C 21 Jul 1999 000654 COBOL II

Chapter 9

Troubleshooting

The following describes typical troubleshooting situations and solutions.

Missing Dump Display Options	124
COBOL Source Display	124
ChangeMan ZMF Integration	125

Missing Dump Display Options

No Debugging and Viewing Server Dumps Available

If the Debug/Viewing Server is not installed at your site, output is written to the //SYSUDUMP DD, or if present, to the //ESPYDUMP DD statement in the execution JCL. You can view Mini Dump reports as part of your job output from the System Display and Search Facility (SDSF).

No Mini Dump Debugging Reports Available

Put a //SYSUDUMP DD statement in the program execution JCL. If you have a //SYSUDUMP DD statement and are not getting Mini Dump report output, contact your installer. Your installer did not set up your prefix.

COBOL Source Display

No COBOL source in Mini Dump or Debug/Viewing Server

If you have a //SYSUDUMP DD statement and are not getting any COBOL source in either the Mini Dump or within the Debug/Viewing Server, the following two scenarios exist:

If this is a production dump:

- 1 You must recompile the program with the correct PROC that contains the Language Server (which stores the source within the Language Data Base) and look at the dump again using the Debug/Viewing Server.
- 2 If you still cannot view the COBOL source, do one of the following:
 - Contact your security administrator and explain that you are supposed to have access to this production program's source code within the Language Data Base.
 - Check if you are using the correct Language Data Base. You can change the Language Data Base you are using on the first Dump/Viewer Server panel by tabbing to Current Source Listing Directory and changing the Language Data Base name.

If this is a testing dump:

- 1 The Repository Data Set (RDS) must contain your COBOL source. If it does not, then recompile your program using the correct options: SOURCE, LIST, MAP, XREF, NOOFFSET. For VS-COBOL, use PMAP instead of LIST and DMAP instead of MAP. The SYSPRINT DD in the compile step must point to the RDS. Also, make sure that you included an //ESPYCOBL DD statement in the execution JCL of the abending program.
- 2 Rerun the test job and look at the dump again using the Debug/Viewing Server (DVS).

- 3 If you still cannot view the COBOL source, your invocation of the Debug/Viewing Server is not allocating your RDS. Contact your installer and explain that you need to have this RDS pre-allocated when you are doing testing/development.

Wrong Version of COBOL Source in the Debug/Viewing Server

If the Debug/Viewing Server is installed and you have chosen to use the Repository Data Set (testing mode) to retain your source COBOL, you must include the following DD statement in the abending program's JCL:

```
//ESPYCOBL DD DSN=xxxxxxxx,DISP=SHR
```

where xxxxxxxx is the name of the partitioned or sequential data set used for your Repository Data Set.

Otherwise, the system assumes that you put the Language Server output to the production Language Data Base.

Failing COBOL Statement Not Displayed

If the program is compiled within ChangeMan ZMF, it must use the LIST and MAP options to see the failing COBOL statement.

ChangeMan ZMF Integration

StarTool DA Fails to Connect with ChangeMan ZMF

If either TCP/IP or the ChangeMan ZMF started task is not already running when you start the Serena® StarTool® DA Workload Server (WLS), [product family] will be unable to connect to ZMF. If this occurs, you should start TCP/IP or ZMF, then issue the following command to initiate a connection:

```
/F ESxxWLS,CMN_SSID=ZMFSubsysID
```

where

ESxxWLS = WLS name assigned during installation of Serena® StarTool® DA

ZMFSubsysID = one-character ID of the particular ChangeMan ZMF subsystem to which you want to connect

To avoid this problem, you should remember to start both TCP/IP and the ZMF subsystem desired before you initialize the Serena® StarTool® DA Workload Server.

Index

A

ABEND 7
Adobe Acrobat 9
after-the-abend compile 70

C

Call Trace Summary
 Debug/Viewing Server 32
 Mini Dump 56, 63
ChangeMan
 extracting source listings 104
ChangeMan ZMF
 history information 103
 information 47, 56
 Information Section 68
 load component information 102
ChangeMan ZMF Data panel 100
ChangeMan ZMF Option 99
COBOL
 debug facilities 14
 processing modes 71
 program listing 42
 SOCB with Language Server 108
 source-level support 71
COBOL Data Section
 Mini Dump 56, 64
commands
 DAS 40
 DISASM 40
 FIND (F) 39
 for Storage Display Panel in Debug/Viewing
 Server 38
 LOCATE (L) 38
 SHOWCOBL 41
compatibility mode 14
compile
 after-the-abend 70
 options required 71
component overview 13
control blocks
 formatted 14

D

DAS command 40

Data Base Index (DBI) 71
Data Management Control Blocks
 Debug/Viewing Server 45
 Mini Dump 56, 66
database support
 Debug/Viewing Server 75, 89
 Mini Dump 56
DB2 dump example 121
DB2 option
 DBRM information 84
 features 15
 host variables 86
 overview 75
 package information 80
 plan information 77
 SQL call summary 76
 SQL statement 86
DB2 options
 DBRM information 15
 host variables 15
 package information 15
 plan information 15
 SQL call summary 15
 SQL statement 15
DBI Selection Panel 18
DBRM information 15
 DB2 option 84
Debug/Viewing Server 42
 Call Trace Summary 32
 COBOL program listing 42
 commands for Storage Display Panel 38
 Data Management Control Blocks 45
 DB2 support 75
 deleting a source listing 22
 described 14
 Diagnostic Section 24
 failing instructions 26
 features 14
 getting started 18
 IMS support 89
 overview 17
 processing a dump data set with IPCS 20
 Program Storage 37
 Register and PSW Information 27
 selecting a dump data set 19
 storage of abending program 38
 System and Job Information 24
deleting a source listing 22
development abends

- language server processing 71
- development mode
 - and language server 71
 - described 12, 71
- Diagnostic Section
 - Debug/Viewing Server 24
 - Mini Dump 58
- diagnostics
 - mini dumps 56
- DISASM command 40
- disassembler function
 - described 14
- dump data set
 - deleting 21
- dump examples
 - DB2 121
 - S130 abend 112
 - S813 abend 115
 - SOC4 - 11 (Storage Not Available) 116
 - SOC4 - 4 (Low Storage Access) 118
 - SOC4 - 4 (Storage Key Mismatch) 119
 - SOC4 - 4 (Storage Protected) 120
- dump types
 - DVS dump 12
 - IBM formatted dump 12
 - mini dump 12
- DVS
 - selecting options 19
 - starting a session 18
 - using the panels 22
- DVS dumps 13

E

- Error Detector Server
 - described 13
- ESPYCOBL DD 71, 124
- extracting ChangeMan source listings 104

F

- failing instructions
 - Debug/Viewing Server 26
 - information about 59
 - mini dumps 56
- FIND (F) command 39

G

- general package information
 - ChangeMan ZMF data 101

H

- history information for ChangeMan ZMF data 103
- host variables 15
 - DB2 option 86

I

- IBM formatted dumps 12
- IMS option
 - features 15
 - IMS region abends 97
 - information displayed 91
 - point-and-shoot feature 94
 - relocated PCBs 96
 - viewing a DVS dump 92
 - what is supported 90
- IPCS
 - default values 20
- IPCS mode 14
- IPCS viewer 14, 20

J

- JCL overrides
 - ESPYCOBL DD 71, 124
- Job and System Section
 - Mini Dump 56, 57

L

- Language Server
 - COBOL & SOCB 108
 - described 14
- language server
 - for development abends 71
 - for production abends 71
- libraries
 - non-linklist and non-LPA 59
- library lookaside 59
- listings, source 104
- LLA 59
- Load a source listing to DA 49
- load component information
 - ChangeMan ZMF data 102
- local dumps 18
- LOCATE (L) command 38
- low storage access
 - SOC4 118

M

Mini Dump

- Call Trace Summary 56, 63
- ChangeMan ZMF Information 68
- COBOL Data Section 56, 64
- Data Management Control Blocks 56, 66
- database support 56
- Debug Report 56
- Diagnostic Section 58
- Job and System Section 56, 57
- PSW and Register Section 56, 61
- Working Storage Section 56, 64

Mini Dump Report sections

- call trace 63
- ChangeMan ZMF Information 68
- COBOL data 64
- data management control blocks 66
- diagnostic 58
- failing instruction 59
- job and system 57
- program status word 61
- registers 61

mini dumps 12

- diagnostics 56
- failing instructions 56, 59

modes

- development 71
- differences between production and development 71
- of operation for COBOL 12
- production 70

multiple viewing options 14

O

online documentation 9

P

package information 15

- DB2 option 80

plan information 15

- DB2 option 77

point-and-shoot facility 14

Primary Option Panel 18

private libraries 59

problems

- solving 123

processing modes

- development 71
- differences between production and development 71
- production 70

production abends

- language server processing 71

production mode

- and language server 70
- described 12, 70

Program Call/Trace Details 49

program listing

- COBOL 42

program status word 61

Program Storage

- Debug/Viewing Server 37

PSW

- see Program Status Word 61

PSW and Register Section

- Mini Dump 56, 61

R

RACF

- security 124

Readme

- accessing 7

Register and PSW Information

- Debug/Viewing Server 27

remote dumps 17, 18, 19

remote viewing 14, 18

repository data set 72

repository data sets 25, 59, 71, 124, 125

S

SDUMP 20

security

- RACF 124

SHOWCOBL command 41

SNAP 20

SOCB

- with Language Server (COBOL II) 108

source listing

- deleting 22

source listings, ChangeMan 104

SQL call summary 15

- DB2 option 76

SQL statement 15

- DB2 option 86

storage key mismatch

- SOC4 119

storage not available

- SOC4 116

storage of abending program

- Debug/Viewing Server 38

storage protected

- SOC4 120

support

information needed 24
System and Job Information
 Debug/Viewing Server 24
SYSUDUMP DD 124

T

troubleshooting
 getting the wrong version of COBOL source
 with DVS 125
 not getting any COBOL source in the Mini
 Dump or DVS 124
 not getting Mini Dump debug reports 124
 not getting the failing COBOL statement in
 the Mini Dump or DVS 125
 using only Mini Dump debug reports 124
types of dumps 12

V

viewer
 IPCS 14, 20

W

Working Storage Section
 Mini Dump 56, 64
Workload Server
 described 14

Z

ZMF Integration Option 7
ZMF option 99