

A Guide for the SAS/C Compiler Consultant

Release 6.50



SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *A Guide for the SAS/C Compiler Consultant, Release 6.50*, Cary, NC: SAS Institute Inc., 1998. pp. 74.

A Guide for the SAS/C Compiler Consultant, Release 6.50

Copyright © 1998 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-133-1

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

Restricted Rights Legend. Software and accompanying documentation are provided to the U.S. government in a transaction subject to the Federal Acquisition Regulations with Restricted Rights. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987). The Contractor/Licenser is SAS Institute Inc., located at SAS Campus Drive, Cary, North Carolina 27513.

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, March 1998

The SAS® System is an integrated system of software providing complete control over data access, management, analysis, and presentation. Base SAS software is the foundation of the SAS System. Products within the SAS System include

SAS/ACCESS®, SAS/AF®, SAS/ASSIST®, SAS/CALC®, SAS/CONNECT®, SAS/CPE®, SAS/DMI®, SAS/EIS®, SAS/ENGLISH™, SAS/ETS®, SAS/FSP®, SAS/GRAPH™, SAS/IML®, SAS/IMS-DL™, SAS/INSIGHT®, SAS/IntrNet™, SAS/LAB™, SAS/MDDB™, SAS/NVISION®, SAS/OR®, SAS/PH-Clinical®, SAS/QC®, SAS/REPLAY-CICS®, SAS/SESSION®, SAS/SHARE®, SAS/STAT®, SAS/TOOLKIT®, SAS/TUTOR®, SAS/Warehouse Administrator™, SAS/DB2™, SAS/GEO™, SAS/GIS®, SAS/SHARE*NET™, SAS/SPECTRAVIEW™, and SAS/SQL-D™ software. Other SAS Institute products are SYSTEM 2000® Data Management Software, with basic SYSTEM 2000, CREATE™, Multi-User™, QueX™, Screen Writer™, and CICS interface software; InfoTap® software; JMP®, JMP IN™, and JMP Serve® software; SAS/RTERM® software; the SAS/C® Compiler; Video Reality™ software; Warehouse Viewer™ software; Budget Vision™, Campaign Vision™, CFO Vision™, Enterprise Miner™, Enterprise Reporter™, HR Vision™, IT Charge Manager™ software, and IT Service Vision™ software; Scalable Performance Data Server™ software; SAS OnlineTutor™ software; and Emulus® software. MultiVendor Architecture™, MVA™, MultiEngine Architecture™, MEA™, and SAS InSchool™ are trademarks of SAS Institute Inc. SAS Institute also offers SAS Consulting® and SAS Video Productions® services. *Authorline*®, *Books by Users*™, *The Encore Series*®, *ExecSolutions*®, *JMPer Cable*®, *Observations*®, *SAS Communications*®, *sas.com*™, *SAS OnlineDoc*™, *SAS Professional Services*™, the *SASware Ballot*®, *SelecText*™, and *Solutions@Work*™ documentation are published by SAS Institute Inc. The SAS Video Productions logo, the Books by Users SAS Institute's Author Service logo, the SAS Online Samples logo, and The Encore Series logo are registered service marks or registered trademarks of SAS Institute Inc. The Helplus logo, the SelecText logo, the Video Reality logo, the Quality Partner logo, the SAS Business Solutions logo, the SAS Rapid Warehousing Program logo, the SAS Publications logo, the Instructor-based Training logo, the Online Training logo, the Trainer's Kit logo, and the Video-based Training logo are service marks or trademarks of SAS Institute Inc. All trademarks above are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

The Institute is a private company devoted to the support and further development of its software and related services.

DB2® and IBM® are registered trademarks or trademarks of International Business Machines Corporation. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

Contents

PART 1 Information for the Compiler Consultant 1

Chapter 1 Δ Service and Support from SAS Institute Inc. 3

Introduction 3

The SAS/C Installation Coordinator and Software Consultant 3

SAS/C Technical Support Service 4

Obtaining Product Updates and Support 9

User Feedback 10

Institute Telephone Numbers 11

Chapter 2 Δ Usage Notes and Zaps 13

Usage Notes Description 13

Usage Notes Tape and Contents 13

Chapter 3 Δ Applying Zaps to the SAS/C Software Products 15

Applying Zaps to the SAS/C Software Products 15

General Information about Zaps and Zapping Utilities 16

Technical Description of APPLYZAP 18

Using APPLYZAP 21

Automatic Application of Maintenance 29

Chapter 4 Δ SAS/C Sample Library 31

Using the Library 31

Chapter 5 Δ Electronic Support Services 33

Introduction 33

Using the SAS Technical Support Web Site 34

Using the File Transfer (FTP) Facility 44

Using E-mail to Report Problems 44

PART 2 Appendices 47

Appendix 1 Δ APPLYZAP Error Messages 49

Return Codes 50

Messages 51

Appendix 2 Δ Using OBJZAP on MVS 63

Introduction 63

Index 69

Information for the Compiler Consultant

- Chapter 1* **Service and Support from SAS Institute Inc.** 3
- Chapter 2* **Usage Notes and Zaps** 13
- Chapter 3* **Applying Zaps to the SAS/C Software Products** 15
- Chapter 4* **SAS/C Sample Library** 31
- Chapter 5* **Electronic Support Services** 33
-

Service and Support from SAS Institute Inc.

<i>Introduction</i>	3
<i>The SAS/C Installation Coordinator and Software Consultant</i>	3
<i>SAS/C Technical Support Service</i>	4
<i>When to Call Technical Support</i>	4
<i>Information Needed by Technical Support</i>	5
<i>Collecting Diagnostic Information</i>	6
<i>Isolating Code Generation or Compiler Problems</i>	6
<i>Isolating Link Problems</i>	6
<i>Isolating Run-time Problems</i>	7
<i>Reading a SAS/C Library Traceback</i>	7
<i>Submitting Diagnostic Information to SAS/C Technical Support</i>	8
<i>Obtaining Product Updates and Support</i>	9
<i>User Feedback</i>	10
<i>Institute Telephone Numbers</i>	11

Introduction

This guide describes the support services provided by SAS Institute for the SAS/C Compiler products. This guide tells you how to report problems, gives information that is useful for diagnosing suspected problems, explains SAS/C Compiler product maintenance procedures, and describes how to use our electronic support facilities.

The last section of this chapter tells you how to reach SAS/C Technical Support. Feel free to contact us with any questions about this guide or about features of the SAS/C and SAS/C++ product suite.

The SAS/C Installation Coordinator and Software Consultant

Licensing SAS/C software includes filling out a Customer Service Information Sheet. The information sheet identifies two persons at your site who will deal directly with SAS/C Technical Support.

The SAS/C Installation Coordinator is the person in your organization responsible for product installation and maintenance. The Installation Coordinator is familiar with the hardware platform and local development environment. SAS Institute ships product updates and installation instructions to the Installation Coordinator.

The SAS/C Software Consultant (sometimes called the Support Representative) is the liaison between SAS/C Technical Support staff and developers using the compiler products at your site. The Software Consultant has an intimate knowledge of the C/C++ language and is familiar with the hardware and software installed at your site.

The designation of an Installation Coordinator and a Software Consultant does not preclude developers at your site from directly contacting SAS/C Technical Support. Experience shows that SAS/C Technical Support can be more effective, and resolve questions more quickly, by working initially through your designated representatives. After a preliminary diagnosis and review of a reported problem with the Software Consultant, SAS/C Technical Support may contact the developer who encountered the problem directly.

SAS/C Technical Support Service

The SAS/C Compiler Research and Development Division provides technical support for the compiler products. We strongly believe that having technical support integrated within the Compiler R&D division ensures that our customers receive support of the highest quality in the industry. SAS/C Technical Support services are provided free of charge to licensed customers.

When contacting the SAS Technical Support Division by phone, identify the question or problem as related to the SAS/C Compiler. SAS/C Technical Support consultants are available weekdays between 9:00 AM and 5:00 PM., eastern standard time.

Guidelines for reporting problems and for gathering information that may be requested by a SAS/C Software consultant follow.

When to Call Technical Support

Some problem types and their symptoms are listed below.

- Compiler-related problems. The compiler:
 - generates an unexpected warning or error for correct source code.
 - fails to accurately diagnose incorrect source code.
 - generates incorrect object code.
 - detects an internal error condition and exits with a CXERR.
 - terminates abnormally, with or without an ABEND code.
- COOL or LKED-related problems. The linker:
 - reports unresolved references.
 - terminates abnormally, with or without an ABEND code.
- Run-time problems. A library or function:
 - does not produce correct results.
 - fails to diagnose erroneous conditions.
 - produces unexpected warning or error messages.

- generates 12xx ABEND (abnormal termination) that is unexplained: for example, not the result of a call to **abort()** or a storage overlay.
- generates an unexplained ABEND.
- after upgrading to a new library version, an existing program fails to execute properly.

Contact the local Software Consultant prior to contacting SAS/C Technical Support. The Software Consultant may be able to quickly resolve a problem. But where assistance is required, the Software Consultant may contact SAS/C Technical Support.

Information Needed by Technical Support

Provide the following information to the Software Consultant before he or she calls to report a SAS/C problem:

- your eight-digit site number. This number can be found on the first page of a compiler listing, and on cover letters accompanying your SAS/C product media. The number also appears on the Institute Program Products Master License agreement.
- the product and version number. For example: SAS/C 6.50C, SAS/C++ 6.0 0C. The version is printed on the first page of the compiler listing and is included on cover letters received with the product. The version of the library can be obtained by using the **=version** run-time option. The version number is of the form *n.nnx*, where *n* is a digit and *x* is an alphabetic suffix.
- operating system and version.
- a phone or fax number or e-mail address, including any special contact instructions such as shift schedules.
- a clear description of the problem, including its external symptoms, the results of your investigation, and the problem type (related to the compiler, the linker, or run time).
- a clear description of the application structure, for example, single or mixed language, all-resident, multitasking, client/server, and so on.
- diagnostic information, including:
 - the *exact* text of any diagnostic messages. Many diagnostic messages contain dynamic substitution of parameters, based upon the existing conditions. It is imperative that we have the complete diagnostic message. This includes messages produced by SAS/C and the operating system.
 - a library *traceback* if the problem involves an ABEND. “Reading a SAS/C Traceback” shows a sample traceback and explains how to interpret it.

- if it can be re-created, a code segment demonstrating the problem in a form that may be transmitted to SAS/C Technical Support.

Collecting Diagnostic Information

There are many diagnostic features within the compiler and run-time library to assist in the isolation of problems, and having this information handy saves time during consultations with Technical Support. The sections below tell you how to enable compiler, linker, and run-time diagnostics.

Isolating Code Generation or Compiler Problems

- Ensure that the compiler option **warn** (enabled by default) is not disabled.
- Enable the **hlist**, **ilist**, and **mlist** options. These options cause the compiler to print, to the listing file, the complete header and macros processed by the preprocessor.
- For code generation problems, enable the OMD (Object Module Dissambler).
- If your site uses the SAS/C Cross-Platform compiler, use the **-v** (verbose) option to force full message output.
- Provide a complete list of compiler options in effect when the problem occurs, for example, **EXTNAME RENT OPTIMIZE DEFINE** (*any_user_defines*). Options affect compiler behavior, and occasionally it is possible to circumvent a problem by specifying different options.
- Provide any traceback generated by the compiler, including the **CXERR** (Compiler Internal Error) message showing the line number and errcode of the error.

Isolating Link Problems

Unresolved external references are the most common problems related to the COOL, ILCLINK, or LKED programs. For all link problems, the following information is required:

- input JCL or Command line syntax
- COOL, ILCLINK, and LKED linkage editor control statements
- expanded JCL or LOG for the command, with any generated error messages and listings
- for COOL, the output generated with the ENXREF option enabled.

Problems reported as link errors sometimes result from source code or compiler details. You should have a detailed knowledge of the source code that generated the input OBJECTS or LOAD MODULEs, and be prepared to discuss the source code and its compilation with a SAS/C Software Consultant.

Isolating Run-time Problems

- For a run-time failure that does not generate a traceback, enable the library run-time options **=btrace** and **=warning**. This will force the library to print all tracebacks and suppressed warnings, assuming the problem has not corrupted library control blocks.
- If a dump is needed and the library is only producing a warning message, use the **=quit**, **=warning**, and **=fdump** run-time options. This will force the library to abnormally terminate after generating the warning message. You can then use the SAS/C Debugger to verify that there are no logic errors within the source code.
- For storage overlays or heap management problems, enable the library run-time option **=storage** to print a storage analysis report at program termination. Then use the SAS/C Debugger, which is very effective at isolating storage problems. For details, see the “storage” command in the *SAS/C Debugger User’s Guide and Reference*.
- Ensure you have complete SAS/C or operating system diagnostic messages. SAS/C diagnostic messages are written to **stderr**, whose exact location is dependent on the operating system and environment: SYSTERM DD (MVS batch), Terminal (VM, TSO), SASE (CICS).

Messages are explained in *SAS/C Software Diagnostic Messages, Release 6.50*. A message consists of an ID and text. The format of a message ID is LSCxnnn, where *x* is C for compiler, D for debugger, G for GENCSEG, I for ILCLINK, L for CLINK, P for the CICS preprocessor, T for the C++ translator, X for the library, or Z for APPLYZAP; *nnn* is the message number. The message number is followed by text. Library messages (prefixed by LSCX) generally have three levels: the first is a severity code; the second is the location of the error; and the third is an explanation. APPLYZAP diagnostic messages are documented in Appendix 1 of this manual.

For operating system problems, the diagnostic message may be one of many types of IBM error messages. Under MVS, these generally start with the letter I. Under VM/SP and VM/XA, the messages may start with DMK (CP) or DMS (VM).

There are some occasions where a message is issued without a number. In these cases, it is even more important to get the entire text of the message. If you are running under VM, check to see if messages are turned on by using the CP command QUERY SET. Turn messages on with the command SET EMSG ON. Under TSO, ensure you are running with PROFILE WTPMSG on.

Diagnostic messages can be output to many places. Places to look are:

- MVS job log (MVS batch)
- SYSTERM DD file (MVS batch)
- terminal (VM, TSO)
- program output file.

Reading a SAS/C Library Traceback

The traceback feature is extremely helpful for isolating the location of an ABEND. Often the traceback contains the exact line that caused the abnormal behavior. Additionally, the traceback shows the calling sequence of active functions, line

numbers, and offsets, at the time of the error conditions. A typical ABEND traceback is shown below, with line numbers added for purposes of reference.

Calling trace:

	Function	Line	Offset	Context
12	LSCABTM(LSCABTM)	---	000124	
11	SIGROUT(LSCZRTE)	--	000228	Raising signal SIGABRT
10	SIGLDR(LSCZLDR)	--	00007A	
9	RAISE(LSCZGEN)	--	00008E	
8	ABORT(LSCABRT)	--	000052	
7	CXERR(FILE1)	865	000152	
6	REPDEL(TASN)	974	0000B2	
5	TREF(TASN)	297	00006C	
4	RVFQ(TASN)	214	0001EE	
3	WTHDR(WTHDR)	149	0003A4	
2	DYNAMN(MAIN1)	366	0009D2	
1	MAIN(LC1370S)	68	000128	

Line 1 indicates that function **main()** of the C MAIN program, in source file **lc1370s**, executed a call to function **wthdr()** at line number 68 and offset 000128. The function call forced SAS/C dynamic loading to occur and subsequently a call to **wthdr()**.

Line 2 indicates that a function **dynamn()** in source file **main1**, at line 366 and offset 0009d2, called the function **wthdr()**.

Lines 3–5 illustrate the calling sequence for functions on the stack. The line numbers and offsets show where function calls were made.

Line 6 indicates that an error probably occurred at line 974 of function **repdel()** in source file **tasn**. **repdel()** called **cxerr()** to generate an ABEND and subsequent compiler messages for a CXERR condition.

Lines 7–12 show the sequence of SAS/C Library functions that get invoked when the function **abort()** is called within a C program.

Submitting Diagnostic Information to SAS/C Technical Support

The previous sections have explained how to obtain detailed descriptive information about a SAS/C problem. Sometimes this is not sufficient. If you contact SAS/C Technical Support, the consultant may request additional information about your problem. Additional information that is often requested and how to obtain it are described below.

- 1 Source code showing function calls, parameter lists, and so on.
- 2 Compiler and linkedit listings and maps.
- 3 Logs of program execution generated with the **=warning** and **=btrace** run-time options enabled. Commands for capturing execution logs are
 - MVS BATCH: enable MSGLEVEL=(1,1).
 - MVS TSO: enable Session Manager, execute the program with the problem, and issue the command SMCOPY TODATASET(*your.dataset*). Session Manager will

then copy the entire contents of the Session Manager window to *yourid.your.dataset*.

- VM CMS: issue the command `SPOOL CONSOLE * STAR`, execute the program with the problem, then issue the command `SPOOL CONSOLE * STOP CLOSE`. The log will be sent to your reader.

4 A system ABEND traceback dump.

On MVS BATCH, issue `//SYSDUMP DD` or `//SYSMDUMP DD`.

Note: If Abend-AID (a dump formatting utility) is installed at your site, turn it off for this job by adding the following statement to your execution JCL:

`//ABNLIGNR DD DUMMY`. But first, check with your systems programmer to ensure that your site does not use a different mechanism for turning off Abend-AID. Δ

On MVS TSO, allocate a `SYSUDUMP` or `SYSABEND` data set, then execute the program with the problem.

On VM CMS, obtain dumps for program exceptions or library-produced abends by doing the following:

- Issue the commands `SET RUN OFF` and `TRACE PROG` before invoking your program. This causes CP to transfer control to you when the exception occurs.
- Issue the command `DUMP T` to send the dump to the printer.
- When the dump is complete, issue the `BEGIN` command. The abend will resume, and a traceback from the library will be produced.
- To turn off `PROG` tracing, issue the `TRACE END` command.
- After a library-produced ABEND (such as an ABEND U1207), spool your console to your reader, then enter the `DEBUG` command. This will produce a register map and other information. Next, close the console log and then obtain a dump in the same way as for a program exception.

Note that a dump from a 2M machine can generate 25K lines to the printer. Make your machine size as small as possible before generating a dump. If you want to send the dump on tape, you can spool your printer to your reader (`SPOOL PRINT *`). Then you can use the `VM MOVEFILE` command to copy the dump to tape. Make sure the input DDname has a blocksize of 133 bytes. Tapes produced by a VM command including `VMFPLC2`, `TAPE`, and `SPTAPE` are acceptable.

See Chapter 5 for instructions on how to get requested information to SAS/C Technical Support.

Obtaining Product Updates and Support

SAS/C Technical Support provides regular maintenance updates for compiler and library products and makes this material available to SAS/C customers in various ways. The Installation Coordinator is responsible for installing and maintaining SAS/C products at your site.

SAS/C problems are tracked by *Usage Notes*. Each Usage Note describes the symptoms of a problem, identifies the component and release where the problem occurs, and tells how to fix or avoid the problem.

The Installation Coordinator corrects software defects by applying zaps to the compiler objects or load modules, using the **APPLYZAP** utility. If a zap is available for a problem, this will be indicated in the Usage Note for that problem. Not every problem described in a Usage Note has a corresponding zap, but every zap is referenced in one or more Usage Notes.

The SAS/C Usage Notes Product Tape contains all Usage Notes and zaps available at the time of the tape's issue. New SAS/C customers receive this tape at time of purchase. SAS/C Technical Support periodically updates the tape, which is shipped automatically to licensed customers.

Chapter 2 contains detailed information about Usage Notes. Chapter 3 explains how to apply zaps to compiler products. Applying zaps is also explained in the installation instructions accompanying the SAS/C Compiler.

Chapter 5 explains how to access the SAS Institute Inc. Web page from a Web browser. From the SAS/C support page, you can search Usage Notes for current information about SAS/C problems and solutions, download zaps, and report problems via an online form.

User Feedback

Beyond Technical Support, there are other mechanisms for providing user feedback. The annual SAS Users Group International (SUGI) conference provides a forum where SAS users may share ideas and suggestions with each other and with Institute staff members. Your suggestions are used as the basis of the SASware Ballot, which is sent as part of the SUGI conference registration packet to all names on the *SAS Communications* mailing list. The results of the ballot are presented at the SUGI conference and are printed along with the proceedings of the conference. The valuable user feedback provided through the SASware Ballot has a major influence on the future direction of all Institute software and services. SUGI is open to all users of the SAS System or SAS/C software.

The SUGI conference includes formal paper presentations, roundtable luncheons, tutorials, demonstrations, and informal "birds-of-a-feather" sessions. Among the conference topics are a number of items of specific interest to users of SAS/C software. The Institute encourages you, as SAS/C software customers, to take an active part in the SUGI conference and welcomes your suggestions for conference topics. Sample topics might include debugging with the SAS/C Debugger, interfacing with other software, portability issues, and specialized areas of support fitting your needs. Please submit your topic suggestions in writing to the Technical Support Division, Attention: SAS/C Software.

Details about upcoming SUGI conferences, as well as the latest information about SAS Institute products and support services, can be found in *SAS Communications*, a quarterly magazine for SAS Institute software users. To add your name to the mailing list, contact the mailing list coordinator at SAS Institute.

The Institute is also interested in assisting SAS/C software customers who would like to start a C user's group. To obtain information about starting a C user's group in your area, contact your Marketing Representative for SAS/C software.

Institute Telephone Numbers

Table 1.1 lists the telephone numbers that the Installation Coordinator, Software Consultant, and user can call to obtain support for the SAS /C product.

Table 1.1 SAS Institute Phone Numbers

Main Phone	(919) 677-8000
Fax	(919) 677-4444
Technical Support Phone ¹	(919) 677-8008
Online Customer Support Facility	(919) 677-8155

¹ For SAS/C Software Consultants or Installation Coordinators only

Usage Notes and Zaps

Usage Notes Description 13

Usage Notes Tape and Contents 13

Usage Notes Description

The Technical Support Division of SAS Institute Inc. provides machine-readable Usage Notes and zaps to assist you in diagnosing and correcting known errors with products. The Usage Notes file is a collection of reports of errors, known problems, information not yet in the reference manuals, and other useful information. The zaps file is a collection of patches to correct problems that are identified in the usage notes.

The Usage Notes are intended to be used as a reference in the event that you encounter a problem with one of the products. If your problem has already been reported, it will be documented in a Usage Note. The Usage Note contains a description of the problem, information about any fixes available, and any cautionary information.

Along with the name of the zap associated with a problem, the Usage Note also specifies the maintenance category of the zap. The maintenance category denotes the severity level of the zap. We advise users to apply all zaps in the **REQUIRED**, **RECOMMENDED**, and **SPECIAL CONSIDERATION** categories. These categories are explained in the section “Usage Notes Tape and Contents”.

Copies of the Usage Notes tape can be requested through the SAS/C Software Technical Support. The Institute requests that you send the tape back to the Distribution Center at SAS Institute after installing the Usage Notes and zap files.

Usage Notes Tape and Contents

The Usage Notes and zaps for SAS/C software products can be obtained for MVS and VM. They are all contained on a magnetic tape with standard labels for all files except those created on VM via VMFPLC2, which creates unlabeled tapes. The tape is either a 1600 BPI, a 6250 BPI, or a 3480 cartridge, depending on your installation requirements. The tape volume serial numbers are of the format UC yy mm , where yy is the year and mm is the month. For example, the Usage Notes tape for August, 1997, has the volume serial UC9708.

Zaps are contained in an MVS PDS, VM MACLIB, or individual files. All zaps should be applied using the APPLYZAP utility for both MVS and VM; however, it is possible to use either OBJZAP or IBM AMASPZAP on MVS and the VM ZAP utility on VM, if necessary.

The members of the zap library consist of individual zaps and zap indices. A zap index contains a list of the zaps that are applicable to a specific release (such as 6.50) and in a specific maintenance category (such as REQUIRED). The names of these index members are of the form \$*rrrxxxx*, where *rrr* are the digits of release number and *xxxx* are the first four characters of the zap category. For example, an index of all the REQUIRED zaps for Release 6.50 is contained in the member \$650REQU. The maintenance categories for zaps are as follows:

REQUIRED

corrects severe problems with incorrect execution, code generation, and so on. This zap should be applied by all sites that license the specified product on the specified operating system.

RECOMMENDED

can be applied without adverse effects at the discretion of the site.

SPECIAL CONSIDERATION

is only for certain situations, or requires special instructions to be applied. This zap requires that you read the corresponding Usage Note before deciding whether to apply it. You should not apply it unless you have encountered the specific situation it describes.

REMOVE

was found to cause problems and should be removed from your system. You should read the corresponding Usage Note for details. Please note that a zap marked REMOVE should only be removed if it has previously been applied to your system.

Applying Zaps to the SAS/C Software Products

<i>Applying Zaps to the SAS/C Software Products</i>	15
<i>General Information about Zaps and Zapping Utilities</i>	16
<i>MVS Zaps</i>	17
<i>VM Zaps</i>	17
<i>Technical Description of APPLYZAP</i>	18
<i>Options</i>	18
<i>Commands</i>	20
<i>Input and Output Files</i>	20
<i>Processing</i>	21
<i>Using APPLYZAP</i>	21
<i>MVS Specifics</i>	22
<i>Example 3.1 Sample MVS JCL to Run to Apply Multiple Zaps to Several Libraries</i>	23
<i>Example 3.2 Sample MVS JCL to Run APPLYZAP to Apply Zaps with the APPLYZAP PROC</i>	24
<i>VM Specifics</i>	25
<i>Example 3.3 VM Sample for Applying Multiple Zaps to Several Libraries</i>	27
<i>Listing Applied Zaps</i>	28
<i>Example 3.4 Sample APPLYZAP Summary Table</i>	28
<i>Automatic Application of Maintenance</i>	29
<i>MVS Automatic Example</i>	29
<i>Example 3.5 Sample MVS JCL for Automatic Application of Maintenance</i>	29
<i>VM Automatic Example</i>	30

Applying Zaps to the SAS/C Software Products

Maintaining SAS/C software products includes applying zaps to one or more of the libraries provided with the products. Utilities available for applying zaps are as follows:

APPLYZAP

A SAS utility for zapping MVS and VM SAS/C libraries, modules, and text files.

AMASPZAP

An IBM utility for zapping MVS load modules.

OBJZAP

A SAS utility for zapping MVS object modules.

ZAP

An IBM utility for zapping VM LOADLIBs, TXTLIBs, and MODULES.

ZAPTEXT

An IBM utility for zapping VM TEXT files.

The recommended utility for applying zaps to SAS/C software products is APPLYZAP. APPLYZAP preprocesses the specified zaps, selects the appropriate zap utilities, applies the zap, records the application of the zap, and produces informational reports (upon request) detailing what zaps have been applied.

General Information about Zaps and Zapping Utilities

In order to facilitate APPLYZAP, zaps provided by SAS Institute Inc. for SAS/C software products contain additional keywords in the NAME statement. The additional arguments are a list of *libnames*, denoting the libraries to which the zap should be applied.

The form of the NAME statement is

```
NAME module-name csect-name libname1 <.ext> <...libname4 <.ext >>
```

The acceptable values of the optional *.ext* argument are shown in Table 3.1.

Table 3.1 Library Filename Extensions

.ext	File Description
.A	SAS/C AR370 archive
.L	MVS load format library or a VM LOADLIB
.M	VM MODULE file
.O	MVS object format library or a VM TXTLIB
.T	VM TEXT file

APPLYZAP uses the list of libnames to determine which libraries the zap is to be applied to (there is a limit of four libnames per zap). Thus, one zap can be applied to several libraries during one execution of APPLYZAP.

Note: If you need to use a zap utility other than APPLYZAP, the libnames will not need to be removed from the NAME statement. The other utilities mentioned previously will ignore everything after the *csect-name* in the NAME statement. Δ

MVS Zaps

Under MVS, zaps include CHECKSUM statements. These CHECKSUM statements are verified by SAS Technical Support before a zap is made available to users. APPLYZAP ensures that the CHECKSUM statements are correct. If CHECKSUM is incorrect, the cause is almost certainly a typographical error. In this case, APPLYZAP prints a diagnostic message and does not apply the zap. Because incorrectly applied zaps may result in incorrect program execution or ABENDs, APPLYZAP ensures that the zaps are applied correctly.

For MVS, zaps also include an IDRDATA statement. The IDRDATA statement is used by APPLYZAP to denote the end of a zap segment and to identify the zap. For each NAME statement, an IDRDATA statement is expected to follow. If a NAME statement is read and an IDRDATA statement is not found before the end of the zap or before another NAME statement is found, an error occurs and processing of that zap stops.

For MVS zaps that apply to load module format libraries, APPLYZAP invokes the IBM utility AMASPZAP. For zaps that apply to object module format libraries (including the AR370 Archived Libraries), APPLYZAP invokes the OBJZAP utility provided by SAS Institute Inc.

VM Zaps

Unlike MVS, VM zaps do not currently include CHECKSUM statements. However, if CHECKSUM statements are provided in the future, APPLYZAP on VM will process them.

VM zaps include a LOG statement that is very similar to the MVS IDRDATA statement. The LOG statement is used by APPLYZAP to denote the end of a zap segment and to identify the zap. For each NAME statement, a LOG statement is expected to follow. If a NAME statement is read and a LOG statement is not found before the end of the zap or before another NAME statement is found, an error occurs and processing of that zap stops.

VM zaps are applied using one of two separate utilities: ZAP or ZAPTEXT. APPLYZAP invokes the ZAP utility to apply zaps to LOADLIBs, TXTLIBs, and MODULES, and it invokes the ZAPTEXT utility to apply zaps to TEXT files and AR370 archived members.

Technical Description of APPLYZAP

The APPLYZAP utility can be invoked with the options shown in Table 3.2. On MVS, the options are specified in the PARM string. On VM, the options are specified as command line options.

Table 3.2 APPLYZAP Options

Options	Negative	Default	Short Form *
CHECK	NOCHECK	CHECK	-C
ECHO	NOECHO	NOECHO	N/A
ERROR	NOERROR	ERROR	-E
FILEDEF	NOFILEDEF	NOFILEDEF	-F <i>VM ONLY</i>
INPUT(zname)	N/A	N/A	N/A
INPUT zname	N/A	N/A	N/A
LINKNAME(lnkn)	N/A	N/A	-Klnkn
LINKNAME lnkn	N/A	N/A	-Klnkn
LIST	NOLIST	LIST	-L
OVERRIDE	NOOVERRIDE	NOOVERRIDE	-O
PRECK	NOPRECK	PRECK	-P
REPORT	NOREPORT	NOREPORT	-R
TERM	NOTERM	TERM	-T
UPPER	NOUPPER	NOUPPER	-U
ZAPNAME(zapu)	N/A	N/A	-Zzapu
ZAPNAME zapu	N/A	N/A	-Zzapu

* Replace the dash (-) with the NOT sign (!) to negate the short form.

Options

CHECK

validates CHECKSUMs.

ECHO

displays diagnostic information about the input read, the output zap statements formed, and the actual command used to apply the zap. There is no short form.

ERROR

instructs APPLYZAP to be tolerant of errors. NOERROR causes termination if any errors occur.

FILEDEF *VM Only*

allows user FILEDEFs to override the *libname<.ext>* keyword.

INPUT(*zname*) || INPUT *zname*

overrides the default input file (MVS: the DDname SYSZIN; VM: the file SYSZIN ZAP), which contains control statements for APPLYZAP. Under MVS, *zname* is searched for in the PDS defined to the SYSZLIB DDname. Under VM, APPLYZAP first checks for a file *zname* on an accessed disk with filetype of ZAP. Then it checks for a member *zname* in a MACLIB with a FILEDEF of SYSZLIB. There is no short form.

Note: Valid APPLYZAP control statements include NAME, VER, REP, IDRDATA, LOG, ALIAS, GO, EXIT, the comment statement (*), the IEBUPDTE control card (./), and the additional APPLYZAP commands as discussed below. △

LINKNAME(*lnkn*) || LINKNAME *lnkn*

overrides the default linkage editor. The default is IEWL for MVS and LKED for VM.

LIST

generates a list of zaps applied and the result of the attempt.

OVERRIDE

instructs APPLYZAP to pass the zap directly to the appropriate zap utility even if APPLYZAP's records indicate that the zap has already been applied.

PRECK

checks for prerequisite zaps.

REPORT

produces a zap information report showing zaps applied during each execution only. Zap informational reports can also be generated for specific libraries using the REPORT command.

TERM

directs diagnostic output to the terminal when running in an interactive session. NOTERM suppresses output to the terminal.

UPPER

causes all output to be uppercased.

ZAPNAME(*zapu*) || ZAPNAME *zapu*

overrides the default utility to be used for applying zaps to MVS LOAD format libraries, VM LOADLIBs, VM TXTLIBs, and VM MODULES. The default utility is AMASPZAP for MVS and ZAP for VM.

Commands

APPLY *zname*

specifies an input file to be preprocessed. Under MVS, *zname* is searched for in the PDS defined to the SYSZLIB DDname. Under VM, APPLYZAP first checks for the file *zname* on an accessed disk with filetype of ZAP, then it checks for a member *zname* in a MACLIB with a FILEDEF of SYSZLIB.

REPORT ZAP=*zname* LIB=*libname*

produces a zap informational report based on the ZAP= and LIB= parameters. Specifying a ZAP=*zname* allows you to narrow the range of a report to only that particular zap. Specifying LIB=*libname* narrows the report to the specified library. To produce an exhaustive list of all zaps that have been applied to a specific library, specify REPORT LIB=<*DDname*>. Either a LIB= or a ZAP= must be specified in each REPORT command.

SYSTEM command

passes a command to the operating system. This command makes use of the C Language **system** function. See the *SAS/C Library Reference, Volume 1, Second Edition* for further details on the **system** function.

Input and Output Files

As shown in Table 3.3, several files are needed during each execution of APPLYZAP. Under MVS, the name of a file is simply a DDname that must be defined in the JCL. Under VM, the name of the file is a filename and a filetype. The only exception to this rule is the SYSZLIB file. SYSZLIB is a DDname under MVS and VM. In this document, a file is referred to only by its DDname or VM Filename.

Table 3.3 APPLYZAP Files

DDname or VM Filename	VM Filetype	File Use
SYSZIN	ZAP	standard input file for APPLYZAP. All APPLYZAP statements are taken from this unless the INPUT(<i>zname</i>) option has been used or an APPLY command is being processed.
SYSZLIB	***	DDname that defines a PDS under MVS or a MACLIB under VM where zaps can be found. This DDname is used for the INPUT option and the APPLY command.
SYSZOUT	LISTING	standard output file for APPLYZAP. All APPLYZAP messages are written to this file.

DDname or VM Filename	VM Filetype	File Use
SYSPRINT	LISTING	standard output file for the zap utilities and the linkage-editor utility.
ZAPPED	MACLIB	log of applied zaps (by module name). This is a PDS under MVS and a MACLIB under VM.
SYSZRPT	LISTING	output from the REPORT option (which produces a list of zaps applied during a run) or REPORT command (which can be used to produce a list of zaps that were previously applied). This is a list of currently applied zaps.
SYSZLIN	TEXT	output of the linkage editor (temporary file).
SYSIN	ZAP	actual zap sent to the zap utility (temporary file).
<i>libname</i>	TXTLIB, TEXT, LOADLIB, MODULE, or A	identifies the SAS/C library containing the member or members to be zapped. Valid library names can be found in the APPLYZAP procedure catalogued under MVS or the APPLYZAP EXEC under VM.

Processing

APPLYZAP processes one zap at a time, each zap having one or more zap segments. A zap segment begins with a NAME statement and ends with an IDRDATA (MVS) or LOG (VM) statement. APPLYZAP validates all CHECKSUMs, ensures that the zap has not been previously applied, ensures that all VER statements verify, ensures that all prerequisite zaps have been applied, applies the zap, generates any necessary diagnostics, and then continues to the next zap. If the zap has already been applied, then APPLYZAP issues a warning and continues to the next zap. If a CHECKSUM does not validate, or if any one of the VER statements does not verify, or if a prerequisite zap has not been applied, then an error is issued and APPLYZAP continues to the next zap (provided that the NOERROR option was not specified).

Zaps are processed regardless of the results of previous zaps, unless a zap utility returns a fatal error or NOERROR is specified. Upon termination, APPLYZAP will generate an appropriate error message and return code.

Using APPLYZAP

The Release 6.50 version of APPLYZAP can be used to apply zaps to any release of the SAS/C software products. The Release 6.50 version of APPLYZAP requires the

Release 6.50 version of the transient library (MVS: sasc.LINKLIB, VM: LSCRTL LOADLIB) to be available. Previous releases of APPLYZAP can be used to maintain SAS/C software of that release and earlier only.

Note: APPLYZAP was unavailable on VM prior to Release 5.00G. Δ

MVS Specifics

Table 3.4 provides a list of required DDnames for an execution of APPLYZAP.

Table 3.4 Required DDnames

DDname	DCB notes and Special Cases
SYSPRINT	LRECL=121, RECFM=FBA
SYSTEM	
SYSZOUT	
SYSZRPT	Only required if the REPORT option or command is used.
SYSIN	
SYSZLIN	LRECL=80, RECFM=FB, BLKSIZE=3200
SYSZIN	Not required if the INPUT option is used (input file must be sequential).
SYSZLIB	Required only if the INPUT option or APPLY commands are used.
ZAPPED	LRECL=132, RECFM=FB
<i>libname</i>	A SAS/C library to be zapped (specified in NAME statement of zap).

For each distinct *libname* found in the NAME statement of a zap being processed, there must be a corresponding DDname. For example, suppose a DDname of LOAD has been defined as follows:

```
//LOAD DD DSN=sasc.LOAD,DISP=SHR
```

Then the following NAME statement can be issued:

```
NAME LC1370 DSUP@: LOAD
```

For a list of possible SAS/C DDnames, refer to the APPLYZAP catalogued procedure.

Example 3.1 illustrates sample MVS JCL to execute APPLYZAP.

Example 3.1 Sample MVS JCL to Run to Apply Multiple Zaps to Several Libraries

```

//jobname JOB job-card information
//APPLYZAP EXEC PGM=APPLYZAP,
//          PARM='UPPER'
//STEPLIB DD DSN=sasc.LOAD,DISP=SHR
//          DD DSN=sasc.LINKLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=121,RECFM=FBA)
//SYSZRPT DD SYSOUT=*
//SYSZLIN DD DSN=&&ZLIN,DISP=NEW,SPACE=(TRK,(2,2)),
//          UNIT=SYSDA,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//SYSZOUT DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//ZAPPED DD DSN=sasc.ZAPPED,DISP=OLD,
//          DCB=(LRECL=132,RECFM=FB)
//SYSIN DD DSN=&&TEMP,DISP=NEW,SPACE=(TRK,(2,2)),
//          UNIT=SYSDA,DCB=(LRECL=80,RECFM=FB)
//SYSZLIB DD DSN=uprefix.SASC.ZAPS,DISP=SHR
//LOAD DD DSN=sasc.LOAD,DISP=SHR
//ARESOBJ DD DSN=sasc.ARESOBJ,DISP=SHR
//LINKLIB DD DSN=sasc.LINKLIB,DISP=SHR
//SYSZIN DD *
*
APPLY Z6001509
*
REPORT ZAP=Z6001509
*
* NAME: Z6001509 PRODUCT: SASC CATEGORY: RECO SYSTEM: MVS
* DATE: 08NOV96 STATUS: DZ+UT USAGE-ID: LIBRARY-C1509
*
* LSCX041, S0C4 ABEND in AMCLOSE, L$CRPWT(L$CRPOS), message LSCX548
*
* NOTE: APPLY TO SASC.LINKLIB (TRANSIENT RUN-TIME LIBRARY IN
* LOAD MODULE FORMAT), AND SASC.ARESOBJ (ALL-RESIDENT
* RUN-TIME LIBRARY IN OBJECT FORMAT) USING THE APPLYZAP
* UTILITY.
* NOTE: RELINK OR ZAP ANY ALL-RESIDENT PROGRAMS WHICH INCLUDE
* THIS MODULE.
* END
NAME L$CRPSO L$CRPOS LINKLIB ARESOBJ
CHECKSUM
VER 0C7A 5850,5000
VER 0F7A 5850,5000
REP 0C7A 0707,0707
REP 0F7A 0707,0707
CHECKSUM CF52D552
IDRDATA Z6001509

```

```

*
REPORT LIB=LOAD
*
REPORT LIB=LINKLIB
/*
//

```

Example 3.1 uses APPLYZAP to apply several zaps to different libraries and also to obtain several zap informational reports. APPLYZAP will decide which utility and library to use for each zap. Note that zaps may be applied using the APPLY command, in which case APPLYZAP looks in SYSZLIB for the zap, or by directly including the zap into SYSZIN.

Example 3.2 illustrates sample MVS JCL to execute APPLYZAP using the APPLYZAP catalogued procedure.

Example 3.2 Sample MVS JCL to Run APPLYZAP to Apply Zaps with the APPLYZAP PROC

```

//jobname JOB job-card information
//APPLYZAP EXEC APPLYZAP,
// PARM.APPLY='UPPER'
//APPLY.SYSZLIB DD DSN=uprefix.SASC.ZAPS,DISP=SHR
//APPLY.SYSZIN DD *
*
APPLY Z6001507
*
REPORT ZAP=Z6001507
*
* NAME: Z6001507 PRODUCT: SASC CATEGORY: RECO SYSTEM: MVS
* DATE: 12DEC96 STATUS: DZ+UT USAGE-ID: CXX-C1507
*
* LSCT439, Object of abstract class <class_name> cannot be declared
*
* NOTE: APPLY TO SASC.LOAD (COMPILER AND UTILITIES LIBRARY IN
* LOAD MODULE FORMAT) USING THE APPLYZAP UTILITY.
* END
NAME LCXX# DECLARE@ LOAD
CHECKSUM
VER 002DEE 4390,F081
VER 002DF2 8990,001B
VER 002DF6 8890,001F,1299
*
REP 002DEE 9110,F081
REP 002DF2 4780,538C
REP 002DF6 D503,8000,C044
CHECKSUM 06C5E186
IDRDATA Z6001507
*

```

```
REPORT LIB=LOAD
/*
//
```

Example 3.2 applies zaps with the APPLYZAP catalogued procedure, which is provided to help in executing APPLYZAP. The APPLYZAP catalogued procedure is located on the SAS/C Usage Notes tape in the .CNTL file. If the APPLYZAP catalogued procedure has been installed into an accessible catalogued procedure library (.PROCLIB) and edited to contain installation-specific names, this example may be used to accomplish the same task as Example 3.1.

In both Example 3.1 and Example 3.2, zap Z6001507 is found in the PDS file defined at DDname SYSZLIB and is contained directly in the JCL file. Listed below are descriptions of the options and commands used with the example APPLYZAP jobs:

REPORT ZAP=*zname*

generates a report containing a list of all the libraries that the named zap (*zname*) has been applied to.

REPORT LIB=*libname*

generates a report containing a list of all the zaps applied to the named library (*libname*).

PARM='UPPER'

generates all output in uppercase.

Use the APPLYZAP catalogued procedure to ensure that all of the necessary data sets are allocated.

APPLYZAP will apply any number of zaps, however, the input file (SYSZIN) must be sequential. Therefore, a specific PDS member as well as concatenated PDS members are acceptable.

VM Specifics

The libname keyword used in the NAME statement of a zap, the LIB= parameter of the REPORT command, and the libname keyword found in all APPLYZAP output, are used to identify a particular VM SAS/C software file.

For each distinct libname found in the NAME statement of the zap being processed, there must be a corresponding VM file or VM FILEDEF pointing to a VM file. In determining how to interpret the libname, APPLYZAP considers the following:

- 1 Was *.ext* supplied?
- 2 Was the FILEDEF option specified?
- 3 Does an actual FILEDEF that matches the libname exist?

Table 3.5 provides a Truth Table that indicates how APPLYZAP will interpret the libname based on the above considerations for a libname of LOAD or LOAD.L.

Table 3.5 APPLYZAP Options

.EXT Supplied?	FILEDEF Option ON?	FILEDEF Exists?	File/DDname to Search for
yes	yes	yes	DDN: LOAD
yes	yes	no	LOAD LOADLIB *
yes	no	yes	LOAD LOADLIB *
yes	no	no	LOAD LOADLIB *
no	yes	yes	DDN: LOAD
no	yes	yes	Error
no	no	yes	DDN: LOAD
no	no	no	Error

For a list of supplied DDnames, refer to the APPLYZAP EXEC.

There are no DDnames required to run APPLYZAP under VM unless a particular zap indicates it by not supplying an *.ext* in the libname. However, it is possible, using the FILEDEF option, to cause APPLYZAP to zap a file pointed to by a FILEDEF that you supply.

Another use of DDnames in APPLYZAP is with the INPUT option and the APPLY command. Zaps specified in the INPUT option and in the APPLY command can be obtained from a MACLIB defined to the DDname of SYSZLIB. It is important to realize that APPLYZAP will first search for the zap on an accessed disk with a filetype of ZAP. By default, the DDname SYSZLIB is allocated to the ZAP MACLIB (the name of the MACLIB on the Usage Notes tape that contains all of the zaps).

To run APPLYZAP on VM, first ensure that the mini-disks containing APPLYZAP and the zaps to be applied are at least read accessible. Next, ensure that the mini-disks containing the libraries to be zapped, the ZAPPED MACLIB (which is created on the first run of APPLYZAP) and LC370 LOADLIB, are write accessible. Then do one of the following:

- Copy the zap to a file with the name of SYSZIN ZAP and then issue the following command:

```
APPLYZAP options...
```

- Make use of the INPUT option by issuing the following command:

```
APPLYZAP INPUT zname additional-options
```

where *zname* is the filename of the zap with a filetype of ZAP, or a member of a MACLIB FILEDEFed to SYSZLIB.

Example 3.3 shows VM sample for applying multiple zaps to several libraries.

Example 3.3 VM Sample for Applying Multiple Zaps to Several Libraries

```

*
*   NAME:Z6001507 PRODUCT:SASC   CATEGORY:RECO  SYSTEM:CMS/ESA
*   DATE:12DEC96  STATUS: DZ+UT  USAGE-ID:CXX-C1507
*
*   LSCT439, Object of abstract class <class_name> cannot be
*   declared
*   NOTE:      APPLY TO CXX LOADLIB (C++ TRANSLATOR) USING
*              THE APPLYZAP UTILITY.
*   NOTE:      IF THE LIBRARY IS INSTALLED IN A DCSS, THE DCSS
*              MUST BE RESAVED.
*   END
NAME      LCXX#      DECLARE@      CXXLOAD
VER       002DEE    4390,F081
VER       002DF2    8990,001B
VER       002DF6    8890,001F,1299
*
REP       002DEE    9110,F081
REP       002DF2    4780,538C
REP       002DF6    D503,8000,C044
LOG Z6001507 ZAPLOG DECLARE@ LSCT439, Object of abstract class
<class_name> cann
*
REPORT LIB=LC370.L

```

Example 3.3 can be applied to the SAS/C software products by following these steps:

- 1 Ensure that all libraries to be zapped are on a write accessible disk. In this case, ensure that the LC370 LOADLIB, the LC370CIC TXTLIB, and the LC370VSE TXTLIB files are on a write accessible disk. The library information can be obtained from reading the header note in each zap; LC370CIC and LC370VSE are associated with zap Z5000158.
- 2 If this is the first time running APPLYZAP, the ZAPPED MACLIB is created on a write accessible disk. If this is not the first time running APPLYZAP, ensure that the ZAPPED MACLIB is on a write accessible disk.
- 3 Ensure that the disk containing the APPLYZAP utility is linked with at least read access.
- 4 If the MACLIB containing the zaps is not named ZAP MACLIB, issue a FILEDEF defining SYSZLIB to the correct MACLIB.
- 5 Invoke APPLYZAP in the following way:

```
APPLYZAP INPUT myzap
```

where *myzap* is the name of a file that must have a filetype of ZAP and contain the data listed in Example 3.3

This example uses APPLYZAP to apply two zaps to different libraries and also to obtain several informational zap reports. APPLYZAP will decide which utility and library to use for each zap.

Note that a zap may be applied using the APPLY command (in which case APPLYZAP will look for a file with a filetype of ZAP on an accessed disk first, then in the MACLIB defined to SYSZLIB for the zap) or by directly including the zap into SYSZIN.

Listing Applied Zaps

APPLYZAP maintains a log of all applied zaps in three ways. First, if the LIST option is in effect, APPLYZAP produces a listing of its run-time activity in the SYSZOUT file. This listing shows which library the zap was applied against and the result of the attempt. If the attempt failed, then the listing will show the reasons for the failure. Regardless of whether LIST or NOLIST is specified, a summary table is generated. This table details the number of libraries used, the number of zaps from input, the number of zaps applied as well as previously applied and not applied, the number of object deck zaps, the number of load module zaps, and the maximum return code from utility. Example 3.4 shows a sample of the APPLYZAP summary table.

Example 3.4 Sample APPLYZAP Summary Table

APPLYZAP Summary:

Zaps:

Name	Applied	NOT Applied	Previously Applied
--	----	-----	-----
Z5500873			X

Statistics:

Number of libraries used	:	1
Number of zaps from input	:	1
Number of zaps applied	:	0
Number of zaps previously applied	:	1
Number of zaps not applied	:	0
Number of object deck zaps	:	0
Number of load module zaps	:	0
Maximum return code from utility	:	4

Second, a log of applied zaps is maintained in the form of a zaps applied file, called ZAPPED. This file, an MVS PDS or VM MACLIB, contains a member for each module zapped. The ZAPPED member-names correspond to the module name. For each zap, information is kept regarding the CSECT name, time, userid, utility used, and filename of the library zapped. This file can be checked to see if a zap has ever been applied.

Third, a report of applied zaps can be obtained by using the REPORT option or the following command: **REPORT LIB=<libname >** .

REPORT processing causes APPLYZAP to list the contents of an internal record of applied zaps for each defined library. The report is written to the SYSZRPT file. A defined library under MVS is any SAS/C library that has been defined with a DD statement. A defined library under VM is a SAS/C library found on a write accessed disk. This method is the most accurate because both the SYSZOUT listing and ZAPPED file can be discarded or overwritten.

Automatic Application of Maintenance

Maintenance to the SAS/C software products is shipped periodically on a Usage Notes tape. This Usage Notes tape contains zaps and Usage Notes. In addition to the zaps themselves, several index files are included. These files are named in the form of \$*rrrxxxx*, where *rrr* are the digits of the release number and *xxxx* are the first four characters of the zap category or the word AUTO (for example, \$650AUTO, \$650RECO, and so on). The \$*rrr*AUTO index file is the focus of this section. Refer to Chapter 2, "Usage Notes and Zaps," for additional information on the other index files.

The \$*rrr*AUTO index file contains several APPLY commands that will cause all of the REQUIRED, RECOMMENDED, and REMOVAL zaps to be applied to the SAS/C software products. Using this file as input to APPLYZAP automatically maintains your SAS/C software. Because APPLYZAP can determine if a zap has been previously applied, attempting to apply a previously applied zap will only result in a descriptive warning. By applying all of the zaps on the current Usage Notes tape, through use of the \$*rrr*AUTO index file, a site can be assured that all of the REQUIRED, RECOMMENDED, and REMOVAL maintenance has been applied to their copy of the SAS/C Software products.

MVS Automatic Example

Examples 3.5 shows sample MVS JCL for automatic application of maintenance.

Example 3.5 Sample MVS JCL for Automatic Application of Maintenance

```
//jobname JOB job-card information
//AUTOZAP EXEC APPLYZAP,
// PARM.APPLY='ECHO'
//APPLY.SYSZLIB DD DSN=uprefix.SASC.ZAPS,DISP=SHR
//APPLY.SYSZIN DD *
*
APPLY $650AUTO
/*
//
```

Example 3.5 applies all of the current REQUIRED, RECOMMENDED, and REMOVAL zaps for Release 6.50 of SAS/C software.

VM Automatic Example

The following sequence of instructions allow all of the current REQUIRED, RECOMMENDED, and REMOVAL zaps for Release 6.50 to be applied to the SAS/C software products.

- 1 Ensure that all of the SAS/C libraries are on a write accessible disk because all zaps will be applied.
- 2 If this is the first time running APPLYZAP, the ZAPPED MACLIB is created on a write accessible disk. If this is not the first time running APPLYZAP, ensure that the ZAPPED MACLIB is on a write accessible disk.
- 3 Ensure that the disk containing the APPLYZAP utility is linked with at least read access.
- 4 If the MACLIB containing the zaps is not named ZAP MACLIB, issue a FILEDEF defining SYSZLIB to the MACLIB.
- 5 Invoke APPLYZAP in the following way:

```
APPLYZAP INPUT $650AUTO ECHO TERM
```

Note: ECHO is specified so that all diagnostics will be generated. NOTERM is specified so that the large amount of diagnostics that are generated with ECHO will not appear on the terminal screen but instead in the SYSZOUT file only. \triangle

SAS/C Sample Library

Using the Library 31

Using the Library

The SAS/C Sample Library is a diverse collection of SAS/C programs included on the Usage Notes tape to assist system and application developers. The library contains programs, functions, and utilities that demonstrate features of the SAS/C Compiler library. The sample library also includes the necessary JCL and directions to compile, link, and execute SAS/C programs. The sample library includes C to assembler and assembler to C programs, interlanguage communication between C and other high-level languages (COBOL, FORTRAN, PL1), TCP/IP client and server applications, along with C and C++ samples. Some of the programs require special products, such as ISPF, DB2[®], or the SAS/C FSSL product in order to execute.

All sample programs contain the following copyright restrictions:

Copyright (c) 1997, SAS Institute Inc.
Unpublished - All Rights Reserved

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina, USA 27513

The table below lists and describes the contents and organization of the sample library:

- The directory names given are the simple directory names on the release media. The full names are dependent on the operating system and the installation.
- Each directory contains a file, ##README, briefly describing the examples in that directory and indicating any special compilation or execution dependencies.
- JCL statements requiring customization for the local environment are flagged by "<==verify" or "<==check." Be sure to verify these statements.

Table 4.1 Organization of the SAS/C Sample Library

Description	MVS Name	CMS Name	UNIX Name
Samples that demonstrate the communication between SAS/C and Assembler	SAMPLE.ASM	SAMPLASM	samples/asm
JCL, sample Debugger configuration files, hypertext cards, ISPF Panel definitions, and the like	SAMPLE.AUX	SAMPLAUX	samples/aux
Sample C program source files	SAMPLE.C	SAMPLC	samples/c
Sample C++ program source files	SAMPLE.CXX	SAMPLCXX	samples/cxx
Header files for the samples	SAMPLE.H	SAMPLHDR	samples/h
Interlanguage Communication Feature samples	SAMPLE.ILC	SAMPLILC	samples/ilc

For each program in the sample library, there is a corresponding file that explains how to compile, link, and execute the program. For example, the FTOC program converts Fahrenheit to Celsius temperature readings. For instructions on compiling, linking, and executing this program:

- on MVS, see **SAMPLE.AUX (FTOC)**
- on CMS, see **SAMPLE MACLIB, MEMBER FTOC.**

Electronic Support Services

<i>Introduction</i>	33
<i>Using the SAS Technical Support Web Site</i>	34
<i>Searching for Information</i>	36
<i>Submitting or Updating a Problem Report</i>	39
<i>Downloading a SAS/C File</i>	42
<i>Using the File Transfer (FTP) Facility</i>	44
<i>Using E-mail to Report Problems</i>	44
<i>Registering for Automatic E-mail Updates</i>	45

Introduction

The three methods of obtaining SAS/C support information or service from the electronic support facilities of SAS Institute Inc. are as follows:

- Using the SAS Technical Support Web Site
- Using the File Transfer (FTP) Facility
- Using E-mail to Report Problems.

All electronic support services require access to the Internet, which is readily available from an Internet Service Provider (ISP). Here is a list of some ISPs likely to provide Internet service for your area:

- CompuServe
- Bitnet
- MCI Mail
- Usenet
- ATTMail
- SprintMail.

Many other ISPs may be available for your area. If your site has a gateway to the Internet, or is on a network with a gateway to the Internet, you can use the SAS electronic support facilities.

To read the SAS Web page, you need a Web browser. While any Web browser may be used, one that supports frames (such as Netscape 2.0 or later or Microsoft Explorer 3.0 or later) is recommended. A Web browser is not required to use the file transfer and e-mail support facilities.

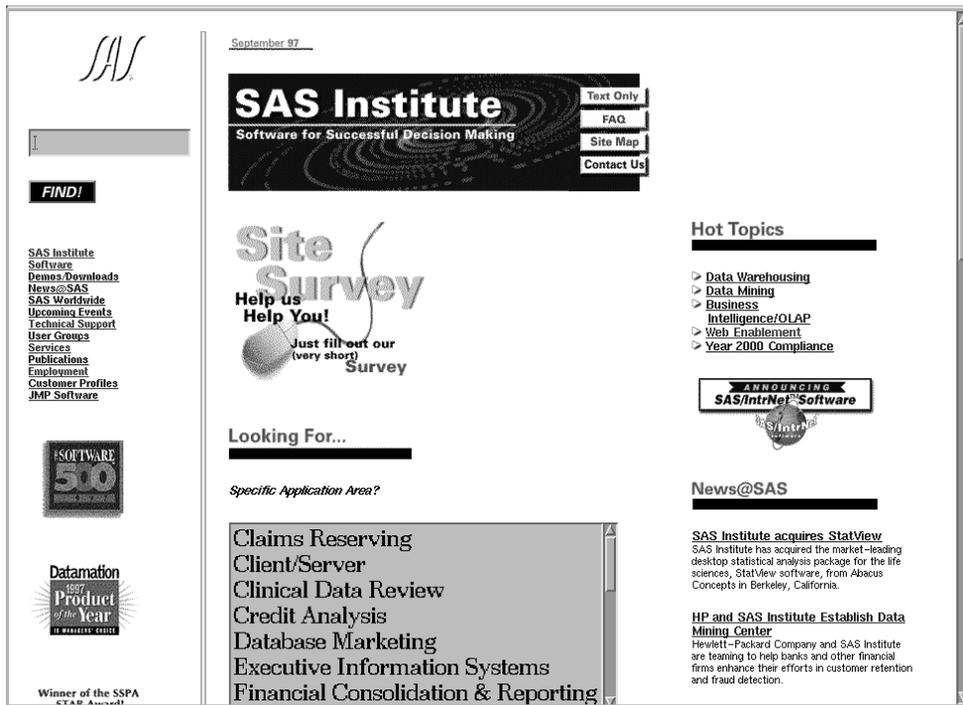
Using the SAS Technical Support Web Site

The SAS Support Web site is the best source of current information about support problems or issues, and it is also the simplest way to obtain SAS Technical Support or information. This section gets you started and discusses the most important SAS/C support services available from the Web page.

The SAS/C Technical Support services appear on the Technical Support main page shown in Figure 5.2 on page 35 . There are two ways to get to this page. You can reach it directly by opening its URL: <http://www.sas.com/service/techsup/intro.html>. Or you can navigate to it from the main SAS Web page as indicated below:

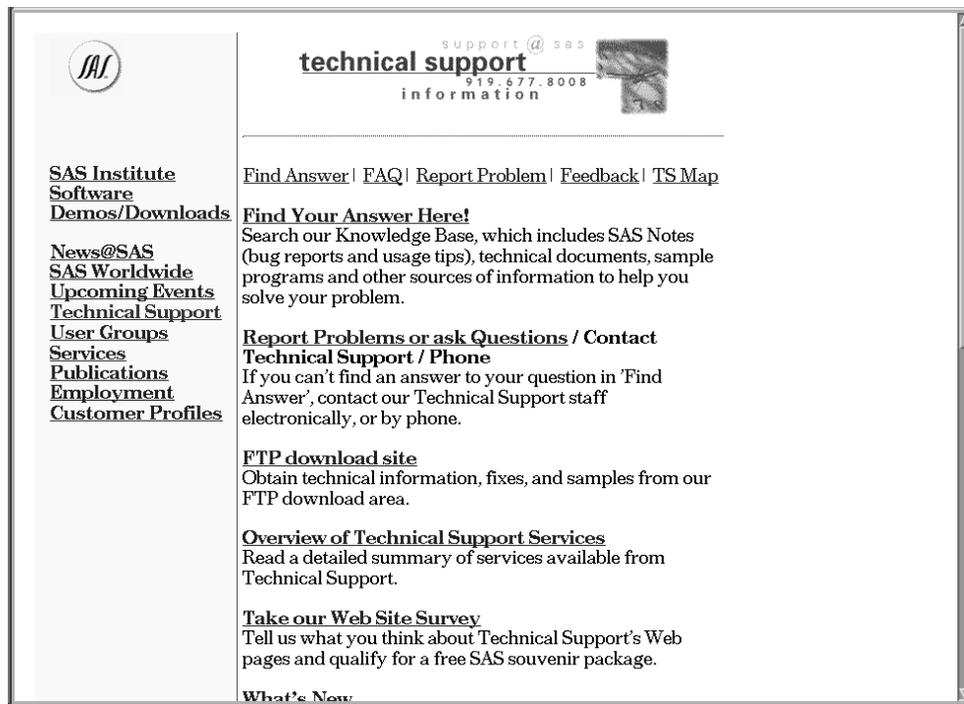
- Open the main SAS Web page by entering its URL: <http://www.sas.com>.

Figure 5.1 SAS Institute Inc. Web Site, Main Page



- From the Main page, click on **Technical Support**: the main Technical Support page appears.

Figure 5.2 SAS Web Site, Technical Support Page



You can reach the SAS/C support and information services from the Technical Support page or you can go to them directly by entering their URLs. The table below lists the main SAS/C support facilities and the URLs for accessing them directly. (Notation `../name.html` indicates that `/name.html` appears at the end of URL <http://www.sas.com/service/techsup>.)

Function	URL
Open the SAS Web site	http://www.sas.com
Open the Technical Support Page	http://www.sas.com/service/techsup
Search for SAS/C information	.../sasnotes.html
Report a SAS/C problem	.../submit_emits2.html
Update a problem report	.../add_emits.html
Designate/display the SAS/C consultant	.../sasrep.html
Update SAS/C consultant information	.../siterep_update.html
Reporting problems using e-mail (explanation)	.../emits.html
Download SAS/C files	.../ftp_sasc.html
Uploading files to SAS/C support (explanation)	.../upload.html

Function	URL
Finding out what's new or changed	../whatsnew.html
Download SAS/C zaps	http://www.sas.com/techsup/download/zaps/sasc
Contacting SAS	http://www.sas.com/contact.html

Note that navigating to the SAS/C support functions, rather than going directly to them, has these advantages:

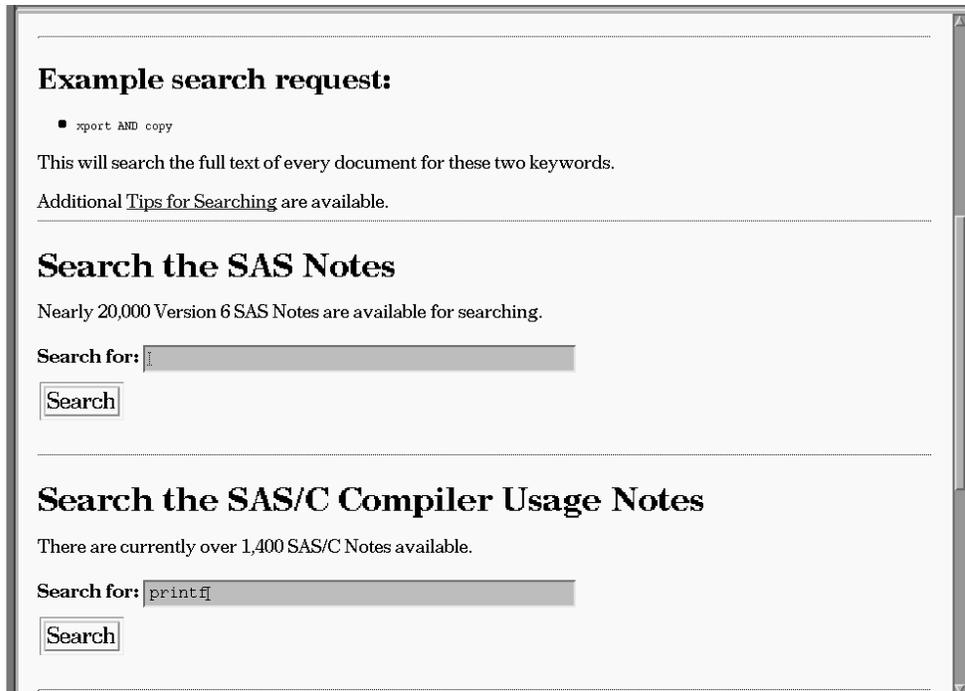
- The URLs are subject to change. If you get an error when attempting to open one of the URLs listed in the table, go to the SAS main page (its URL will not change), navigate to the Technical Support page, and select the desired task or facility from there.
- The Web page is also subject to change. Since the last time you requested SAS/C support or information, an important new feature or piece of relevant information may have been added to the SAS Web site.

We recommend that you periodically browse our Web site, which is continually being enhanced and expanded.

Searching for Information

To search the Technical Support database for information about SAS/C problems and solutions:

- Go to the Technical Support Search form (<http://www.sas.com/service/techsup/sasnotes.html>):

Figure 5.3 Searching the SAS Web Site Databases

Example search request:

- xport AND copy

This will search the full text of every document for these two keywords.

Additional [Tips for Searching](#) are available.

Search the SAS Notes

Nearly 20,000 Version 6 SAS Notes are available for searching.

Search for:

Search the SAS/C Compiler Usage Notes

There are currently over 1,400 SAS/C Notes available.

Search for:

- Scroll to the section “Search the SAS/C Compiler Notes”. In the **Search for** field, enter the keyword or string you wish to search for and click on the **Search** button. For example, searching for “printf” produces the results shown in Figure 5.4.

Figure 5.4 SAS Web Site, Search Results Page


Software for Successful Decision Making

Search Interface

This is a searchable index of information.

Enter keyword(s):

Select an index to search:

Here is the result of your search using the keyword(s) "printf":

- 1: [CENTRY Assembler pgms using stdout may lose output if CEXIT used](#)
Score: 95, Lines: 60, Size: 2 kbytes, Type: HTML file
- 2: [DLI sample corrections for CICS User's Guide](#)
Score: 93, Lines: 55, Size: 1 kbytes, Type: HTML file
- 3: [Bad code is generated for if-statements related to pointer subtraction](#)
Score: 93, Lines: 58, Size: 1 kbytes, Type: HTML file
- 4: [Incorrect generated code for constant sequence 0.0, 1.0, 0.0](#)
Score: 91, Lines: 64, Size: 1 kbytes, Type: HTML file
- 5: [Incorrect generated code for constant sequence 0.0, 1.0, 0.0](#)
Score: 91, Lines: 64, Size: 1 kbytes, Type: HTML file
- 6: [Incorrect results from short to float conversion](#)
Score: 90, Lines: 53, Size: 1 kbytes, Type: HTML file
- 7: [Incorrect results from short to float conversion](#)

- Browse through the list of matches for your topic. When you see an item about which you would like additional information, click on the item. For example, if you click on item 3, “Bad code is generated for if-statements related to pointer subtraction”, a full description of this problem and its solution appears, as shown in Figure 5.5.

Figure 5.5 SAS Web Site, Detailed Search Results Page

OUTSTANDING PROBLEM

COMPILER-C0133

Bad code is generated for if-statements related to pointer subtraction

Products

- SASC

Incorrect object code can be generated for if-statements which test the results of pointer subtractions involving auto variables. For example:

```
typedef struct {unsigned a; unsigned b; unsigned c;} X;
X y(110);
void main() {
  int n;
  n = &y(16) - &y(31); /* n will be 3!! */
  if (n != 0) printf("n is not zero\n");
}
```

The printf statement will not be executed due to invalid code generation.

To correct the problem, apply the zap referenced by 24500133.

Keywords

- QPTR
- REL450F
- POINTER
- SUBTRACTION

Submitting or Updating a Problem Report

SAS Technical Support provides two ways for SAS Software Consultants to report problems electronically:

- the Web problem report form
- the Electronic Mail Interface to Technical Support (EMITS) facility.

The preferred method of reporting a problem is via the Web form. If you want to use e-mail to report your problem, see the directions in “Using E-mail to Report Problems”.

To report a new problem:

- Go to the problem report form (http://www.sas.com/service/techsup/submit_emits2.html).

Figure 5.6 SAS Web Site, Problem Report Page

[TS Home](#) | [Find Answer](#) | [FAQ](#) | [Report Problem](#) | [Feedback](#) | [TS Map](#)

Submit Problems or Questions to Technical Support

ANY user can submit problems or questions to Technical Support via this interface.

When you submit this request, an e-mail message will be sent to the Technical Support Division at SAS Institute. Technical Support will normally respond via e-mail.

NOTE: This page uses Javascript. Some browsers may have trouble running the Javascript code on this page. If you have any problems submitting this form you can use a [non-Javascript coded page](#) to submit your problem.

All fields are required so please fill each of them out correctly.

Your E-mail Address (accuracy is essential):

Your Name:

Name of Your Company:

Phone number (including area code or country code):

Site Number (8 or 10 digit number)
 If you do not have one leave this field filled with zeros:

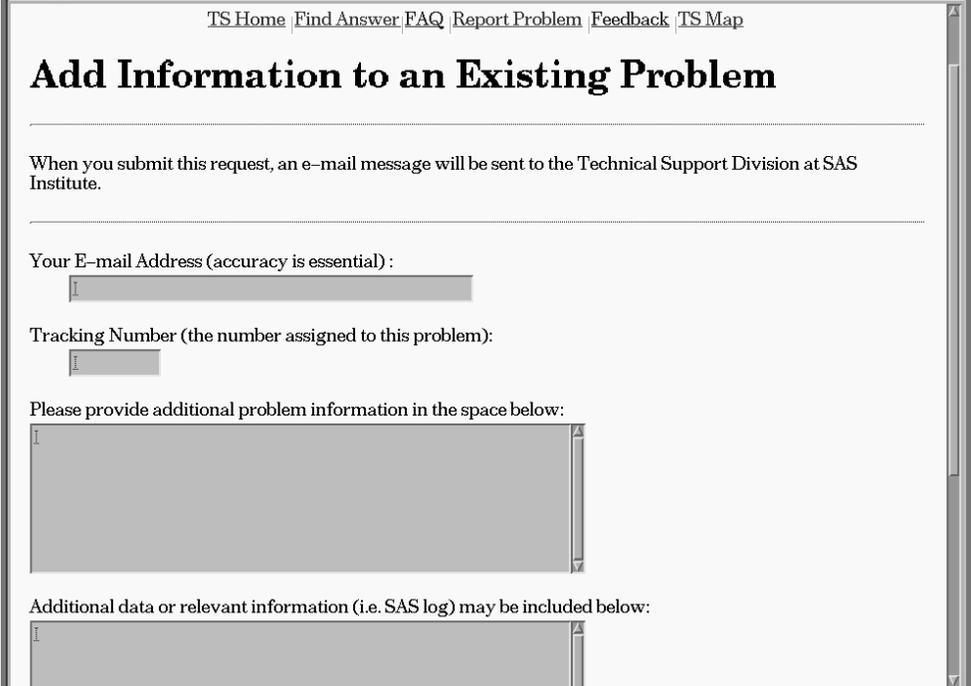
- Fill in, or select, the requested items of information: e-mail address, name, company name, phone number, SAS site number, problem area (select SAS/C), SAS/C release, problem description, and additional relevant diagnostic information.
- Click on **Submit this request** to send or **Clear the form** to cancel or re-enter all information.

You will receive a reply (sent to the e-mail address you specified) acknowledging your submission. Your request is assigned a tracking number.

To update a reported problem:

- Go to the problem update form (http://www.sas.com/service/techsup/add_emits.html).

Figure 5.7 SAS Web Site, Problem Update Page



The screenshot shows a web page titled "Add Information to an Existing Problem". At the top, there are navigation links: [TS Home](#), [Find Answer](#), [FAQ](#), [Report Problem](#), [Feedback](#), and [TS Map](#). Below the title, a horizontal line is followed by the text: "When you submit this request, an e-mail message will be sent to the Technical Support Division at SAS Institute." Another horizontal line follows. The form contains three input fields: 1. "Your E-mail Address (accuracy is essential):" with a text input field. 2. "Tracking Number (the number assigned to this problem):" with a text input field. 3. "Please provide additional problem information in the space below:" with a large text area. Below this is the text "Additional data or relevant information (i.e. SAS log) may be included below:" followed by another text area. A vertical scrollbar is visible on the right side of the page.

- Fill in the requested items of information: e-mail address, tracking number, and additional relevant descriptive or diagnostic information.

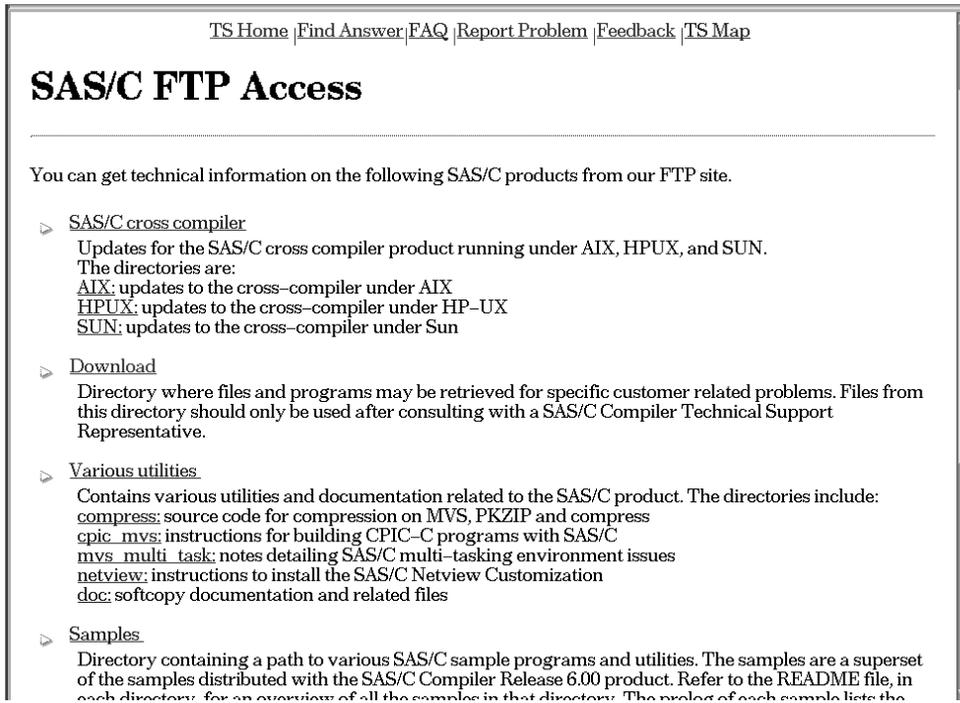
- Click on **Submit this request** to send or **Clear the form** to cancel or re-enter all information.

Downloading a SAS/C File

To download a SAS/C file from the SAS Web site:

- Go to the SAS/C area of the FTP site (http://www.sas.com/service/techsup/ftp_sasc.html).

Figure 5.8 SAS FTP Site, SAS/C Area



- Select the type of file you want to download. For example, to download a SAS/C file, click on **Download**, and a list of available SAS/C files appears.

Figure 5.9 SAS FTP site, SAS/C Download Area

Name	Last modified	Size	Description
Parent Directory	09-Aug-97 16:07	-	
gosobj_lscbrtg	09-Aug-97 16:46	3k	
osiel_jcl	09-Aug-97 16:45	4k	
sasc_c600f_load_seq	09-Aug-97 16:46	1M	
srapstuff.zip	09-Aug-97 16:45	323k	
trace_full.prn	08-Sep-97 18:48	1M	
trace_sum.prn	08-Sep-97 18:48	66k	
transient_load_seq	09-Aug-97 16:46	1M	
treset.text	15-Aug-97 15:47	1k	
zmit	11-Sep-97 14:48	3M	

- Click on the file that you want to download.
What happens when a file you click on to download is platform-dependent? You may be asked to enter or verify the pathname on the local disk for the downloaded file. Or the file may appear in your browser's display area: if this happens, use File->Save to specify the local filename for the downloaded file.

To download a SAS/C zap:

- Go to the SAS/C zap download area (<http://www.sas.com/techsup/download/zaps/sasc>).
- Select the operating system that your SAS/C compiler runs under.
- Select the revision level of your SAS/C compiler.
- Select the zap file you want to download.

If your browser has an upload feature, you can use it to send files to the SAS FTP site. For example, if your browser is Netscape 2.0 or later, you can:

- Open the SAS FTP upload location (<http://www.sas.com/techsup/upload/>). Note that the concluding slash (/) is required.
- Select File->Upload File ...
- A file selection dialog appears. Browse and select the file you want to upload.
Note: Files that you upload to SAS should begin with the letter T followed by your tracking number. For example:

T1234567_open_dump.zip

△

If your browser does not have an upload feature, use either one of several tools available from the Web FTP, as explained in the next section.

Using the File Transfer (FTP) Facility

You can also transfer files between your site and the SAS support facility by directly using a local FTP client program:

- Go to the directory or location on the local host to which you want the downloaded files to be stored.
- Connect to the SAS Anonymous FTP host by typing this command:

```
ftp ftp.sas.com
```

- When prompted for a username, enter ANONYMOUS. When prompted for a password, enter your e-mail address.
- Use the FTP program's **cd** (change directories), **ls** (list files), and **pwd** (display the name of the present working directory) commands to navigate to the SAS/C download area that you are interested in. For example, to go to the SAS/C download directory, display the files there, and download the file named **xmit** by typing these commands:

```
cd /techsup/download/SASC/download
ls
get xmit
```

To download zap **z6001555** for Release 6.00 of the SAS/C compiler for the MVS operating system, type these commands:

```
cd /techsup/download/zap/sasc/mvs/600
get z6001555
```

You can also transfer files from your local system to the SAS FTP site. For example, to transfer a file named **T1234567_open_dump** located in directory **outgoing** to the **upload** directory on the SAS FTP host, type this series of commands:

```
cd outgoing
ftp ftp.sas.com
username: anonymous
password: your_email_address
cd upload
put T1234567_open_dump
```

Using E-mail to Report Problems

Use e-mail to report SAS/C problems or to communicate with SAS/C support consultants. You can also register to receive automatic e-mail updates from SAS support about problems and solutions.

The Electronic Mail Interface to Technical Support (EMITS) system logs your mail to the attention of the Technical Support Division and sends an e-mail message acknowledging receipt of your mail.

You must register to use EMITS. If you have accounts on multiple machines, register from the same host system that you want to use for sending messages to the Technical Support Division. EMITS extracts your e-mail address from the header of your message and uses this address as your account ID. You may register from multiple machines if you want to send messages to Technical Support from more than one machine. The SAS Institute Technical Support e-mail address is SUPPORT@SAS.COM.

In the subject field of the message, type EMITS.

Start in column 1 in the body of the message and type:

```

\./register= customer name
\./company= company name
\./site= valid site number
\./phone= phone number

```

When your account is added, EMITS notifies you by e-mail and sends you a complete guide to using EMITS.

Once you are registered for EMITS, you can receive online documentation by sending an e-mail message with the following test:

```
\./help
```

Here are several points to keep in mind when sending mail to EMITS:

- All e-mail messages to EMITS should be sent to SUPPORT@SAS.COM.
- All e-mail messages intended for EMITS must contain the words EMITS in the subject line.
- EMITS is not case sensitive.
- EMITS can process only one request for each e-mail message that you send. For example, send two separate e-mail messages to SUPPORT@SAS.COM to register and to request the help file.

Registering for Automatic E-mail Updates

Use the Technical Support News Listing (TSNEWS-L) facility to receive automatically technical information such as

- Problem Alert Letters
- Alert Notes.

You can subscribe to TSNEWS-L by sending e-mail to LISTSERV@VM.SAS.COM. The body of your message should consist of this line: SUBSCRIBE TSNEWS-L **your name**.

You can find what is available through TSNEWS-L by sending e-mail to LISTSERV@VM.SAS.COM with this in the body of the message: INDEX TSNEWS-L.

TSNEWS-L responds with a message listing all the documents that are available for downloading. Once you know the name and type of a document you want, issue this command to receive it: GET **filename filetype** TSNEWS-L.

To receive documentation on commonly used LISTSERV commands, send e-mail to LISTSERV@VM.SAS.COM with this in the body of the message: HELP.

P A R T

2

Appendices

Appendix 1 **APPLYZAP Error Messages** 49

Appendix 2 **Using OBJZAP on MVS** 63

APPENDIX

1**APPLYZAP Error Messages***Return Codes* 50*Messages* 51

<i>LSCZ200</i>	51
<i>LSCZ201</i>	51
<i>LSCZ202</i>	51
<i>LSCZ203</i>	51
<i>LSCZ204</i>	51
<i>LSCZ205</i>	52
<i>LSCZ206</i>	52
<i>LSCZ207</i>	52
<i>LSCZ208</i>	52
<i>LSCZ209</i>	52
<i>LSCZ210</i>	52
<i>LSCZ211</i>	53
<i>LSCZ212</i>	53
<i>LSCZ213</i>	53
<i>LSCZ214</i>	53
<i>LSCZ215</i>	53
<i>LSCZ216</i>	53
<i>LSCZ217</i>	54
<i>LSCZ218</i>	54
<i>LSCZ219</i>	54
<i>LSCZ220</i>	54
<i>LSCZ221</i>	54
<i>LSCZ222</i>	54
<i>LSCZ223</i>	55
<i>LSCZ224</i>	55
<i>LSCZ225</i>	55
<i>LSCZ226</i>	55
<i>LSCZ227</i>	55
<i>LSCZ228</i>	56
<i>LSCZ229</i>	56
<i>LSCZ230</i>	56
<i>LSCZ231</i>	56
<i>LSCZ232</i>	56
<i>LSCZ233</i>	56
<i>LSCZ234</i>	57
<i>LSCZ235</i>	57
<i>LSCZ236</i>	57
<i>LSCZ237</i>	57

<i>LSCZ238</i>	57
<i>LSCZ240</i>	58
<i>LSCZ241</i>	58
<i>LSCZ242</i>	58
<i>LSCZ243</i>	58
<i>LSCZ244</i>	58
<i>LSCZ245</i>	58
<i>LSCZ246</i>	59
<i>LSCZ247</i>	59
<i>LSCZ248</i>	59
<i>LSCZ249</i>	59
<i>LSCZ250</i>	59
<i>LSCZ251</i>	60
<i>LSCZ252</i>	60
<i>LSCZ253</i>	60
<i>LSCZ254</i>	60
<i>LSCZ255</i>	60
<i>LSCZ256</i>	61
<i>LSCZ258</i>	61
<i>LSCZ259</i>	61
<i>LSCZ260</i>	61
<i>LSCZ261</i>	61
<i>References</i>	62

Return Codes

APPLYZAP may return the following return codes:

APPLYZAP Return Codes

Return Code	Meaning
0	no diagnostics generated.
4	warning message generated. Refer to SYSZOUT and SYSPRINT for details.
8-24	error messages generated by APPLYZAP or the zap utility. Refer to SYSZOUT and SYSPRINT for details.
1012	termination requested regardless of severity of error. See the SYSZOUT listing for details. Note: This code can be returned only when the NOERROR option is specified.
1016	fatal error in APPLYZAP. See the SYSZOUT listing for details.
1020	internal error in APPLYZAP. Call SAS Institute Technical Support for assistance.

Messages

The following is a list of notes, warnings, and errors that may be issued by APPLYZAP:

LSCZ200

Zap <zap-name> to module <module-name> in <library-name> applied successfully.

Level: NOTE

ZAP applied successfully.

LSCZ201

Zap <zap-name> to module <module-name> in <library-name> applied but generated warnings.

Level: NOTE

ZAP applied successfully, however, it generated warnings. See the ZAP output in SYSPRINT for more details.

LSCZ202

Zap <zap-name> to module <module-name> in <library-name> did not apply.

Level: NOTE

ZAP did not apply. See ZAP output in SYSPRINT for more details.

LSCZ203

Zap <zap-name> to module <module-name> in <library-name> returned a severe error.

Level: NOTE

A severe error occurred in attempting to apply the ZAP. APPLYZAP will not continue. See ZAP output in SYSPRINT for more details.

LSCZ204

Severe error returned from <utility-name> attempting to apply <zap-name> to <module-name> : <library-name>.

Level: NOTE

The utility returned a severe error. Under MVS, see the job log for more details. Under VM, type HELP on any generated VM message for further details. Also, check SYSPRINT.

LSCZ205

Severe error: <utility-name> RC=nn.

Level: ERROR

Ensure that all zapping utilities needed (that is, under MVS, OBJZAP, and AMASPZAP, and under VM, ZAP and ZAPTEXT) are accessible to the program. Also, check SYSPRINT for a more complete explanation of why the ZAP utility terminated.

LSCZ206

Fatal error: <utility-name> RC=nn.

Level: ERROR

Check SYSPRINT for a more complete explanation of why the ZAP utility terminated.

LSCZ207

No module name on NAME card - execution cancelled.

Level: ERROR

A NAME card did not contain a module name. Check NAME statements to find the problem.

LSCZ208

SYSPRINT not defined or LRECL not 121; or RECFM not FBA.

Level: ERROR

An MVS only error. The SYSPRINT file is used by all zapping utilities. This DDname must be defined and AMASPZAP requires it to have an LRECL of 121.

LSCZ209

Zap file write error or out of space - terminating.

Level: ERROR

APPLYZAP could not write to the SYSIN file or not enough space was allocated. Check to ensure proper space allocation and access authority.

LSCZ210

Required ddname not defined or defined incorrectly: <DDname>.

Level: ERROR

An MVS only error. The <DDname> is not defined to APPLYZAP. See instructions for required DDnames.

LSCZ211

Work file cannot not be created; RC=nn.

Level: ERROR

A temporary work file that is used by APPLYZAP could not be created.

LSCZ212

Missing IDRDATA/LOG card - Processing terminated.

Level: ERROR

An IDRDATA or LOG statement was expected and not received. Processing was terminated. Each zap segment begins with a NAME statement and must end with an IDRDATA or LOG statement. A zap segment has been found that does not contain an ending IDRDATA or LOG statement. Check the zap for errors.

LSCZ213

Requested Option <option>.

Level: NOTE

An option was requested at run time. The <option> confirms which option was requested.

LSCZ214

Library specified in zap does not exist.

Level: ERROR

The NAME statement specified a library that is not valid or was not defined. Check the list printed out with the error message for valid libnames and then (MVS) compare that with the JCL or (VM) ensure that the library is on a write accessed disk.

LSCZ215

Invalid checksum characters; must be 8 hex digits <checksum-value-specified>.

Level: WARNING

CHECKSUM specified is not valid. Verify CHECKSUM value given for correctness.

LSCZ216

Checksum error - zap cancelled: <module-name>.

Level: ERROR

CHECKSUM value given was incorrect for zap. Make sure all CHECKSUM statements are in the correct place and have correct values.

LSCZ217

Invalid VER/REP offset specified: <VER/REP statement>.

Level: ERROR

Offset value given on VER or REP card is invalid. Make sure value is correct.

LSCZ218

VER/REP data rejected - invalid: <VER/REP data>.

Level: ERROR

Data value given on VER or REP card is invalid. Make sure value is correct.

LSCZ219

VER/REP data rejected - odd number of digits:

<VER/REP data>.

Level: ERROR

Data value must be given with an even number of hex digits. Make sure value is correct.

LSCZ220

No VER/REP offset found - zap cancelled:

<VER/REP statement>.

Level: ERROR

VER or REP card did not contain an offset value. Check VER/REP statement.

LSCZ221

No VER/REP data found - zap cancelled:

<VER/REP statement>.

Level: ERROR

VER or REP card did not contain a data value. Check VER/REP statement.

LSCZ222

Remaining zap statements discarded due to EOF.

Level: ERROR

A zap statement was found before a valid NAME card was encountered. Make sure zap is specified correctly.

LSCZ223

<ZAPPED | ZAPPED MACLIB> not defined.

Level: ERROR

The zaps applied data set, is not defined. This data set must be defined in order for APPLYZAP to execute. On MVS, issue a DD statement in the JCL for ZAPPED (see JCL examples in Chapter 3). On VM, link the disk that contains the ZAPPED MACLIB as write access.

LSCZ224

Cannot update <ZAPPED | ZAPPED MACLIB> information: <member-name>.

Level: ERROR

An error occurred when attempting to update the ZAPPED information. Check SYSTERM for more details.

LSCZ225

No Memory available - terminating.

Level: ERROR

An attempt to allocate needed memory failed. 4M is the suggested region size under MVS and virtual machine size under VM. Check region specification or machine size.

LSCZ226

Writing <ZAPPED | ZAPPED MACLIB> information:

<member-name>.

Level: ERROR

An I/O error occurred on the zaps applied data set. On MVS, this may indicate a need to compress the data set.

LSCZ227

No valid library specified - see APPLYZAP doc for valid libraries.

Level: ERROR

There are no SAS/C libraries defined. On MVS, define the SAS/C libraries with DDnames LOAD, LINKLIB, and so on. On VM, define the SAS/C libraries by

ensuring that the files are on a write accessible disk. See the USING APPLYZAP section under your specific system for valid library names.

LSCZ228

Module does not exist in library: <library-name>.

Level: ERROR

Module found on NAME card does not exist in the specified library. Ensure that the correct libraries are defined.

LSCZ229

Diagnostics generated: <utility-name> RC=nn.

Level: WARNING

The specified zap may not have applied. Check the return code given by the zap utility and look at SYSPRINT to obtain an explanation of the problem.

LSCZ230

No library name and AUTO not specified.

Level: ERROR

The NAME statement contains no library specification and NOAUTO is in effect. APPLYZAP cannot apply this zap.

LSCZ231

Current zap may not be completely applied.

Level: WARNING

The zap failed to apply to one of multiple libraries; therefore further execution with this zap is terminated. Check SYSPRINT for details on why the zap failed to apply.

LSCZ232

Error in creation of <ZAPPED | ZAPPED MACLIB>.

Level: ERROR

Creating the zaps applied data set produced an error. Under MVS ensure that the DD statement for ZAPPED is correct (see previous JCL examples) and retry. Under VM, ensure that adequate disk space is available and retry.

LSCZ233

No more ZAPPED information kept.

Level: WARNING

Issued to inform the user that for the remainder of this execution, APPLYZAP will not attempt to keep any information in the zaps applied data set. This is a direct result of LSCZ232. See that message for more details.

LSCZ234

Error applying zap - additional libraries will be bypassed.

Level: WARNING

Issued to inform the user that additional libraries may be bypassed due to a previous error. This is a direct result of LSCZ231. See that message for more details.

LSCZ235

Permanent ZAPPED information cannot be obtained - terminating.

Level: ERROR

The specified library had no permanent ZAPPED information when it should have or the information could not be obtained. See SYSTERM and SYSPRINT for more details.

LSCZ236

Zap name incorrect - check IDRDATA or LOG card - <zapname>.

Level: ERROR

The specified zap name was not correct. It should have the form of "trrrxxxx" where rrr is the release, xxxx is a number from 0000-9999 indicating the zap number, and t is "Z", "R", or "\$". Correct the name in the IDRDATA or LOG card and retry.

LSCZ237

Zap already applied - <zapname | Overridden>.

Level: WARNING

The zap has been previously applied. This message may be issued in two separate cases. If an attempt is made to apply a zap that has been previously applied, this message will be issued and processing of this zap will end unless the OVERRIDE option was specified. If a zap requires a prerequisite zap and the prerequisite zap has been previously applied, then this message will be issued and processing of the zap will continue.

LSCZ238

Prerequisite zap has not been applied - <prereq zapname>.

Level: ERROR

A prerequisite zap, required by the current zap, has not been previously applied. Processing of this zap will end unless the NOPRECK option was specified.

LSCZ240

No memory available from POOL - terminating.

Level: ERROR

The initial attempt to allocate memory failed. Increase region and retry.

LSCZ241

Attempt to bootstrap OBJECT module information failed - <SYSZLIN | SYSZLIN TEXT | library-name>.

Level: ERROR

APPLYZAP attempted to create the initial permanent ZAPPED information for the object format library <library-name>. See SYSTERM and SYSPRINT for more details.

LSCZ242

Attempt to bootstrap LOAD module information failed - <library-name>.

Level: ERROR

APPLYZAP attempted to create the initial permanent ZAPPED information for the load format library <library-name>. See SYSTERM and SYSPRINT for more details.

LSCZ243

Attempt to apply permanent zap information failed - <utility-name> RC=nn.

Level: ERROR

APPLYZAP attempted to maintain the permanent zap information and failed. See SYSTERM and SYSPRINT for more details.

LSCZ244

Load module zapped information failure - <library-name>.

Level: ERROR

An attempt was made to retrieve the permanent zap information and failed. For more details on the cause of the failure, see SYSTERM and SYSPRINT.

LSCZ245

Processing of zap terminated due to application failure - <zap-name>.

Level: ERROR

One or more VER statements did not verify; therefore, a failure resulted. Either the zap has been previously applied using another utility besides APPLYZAP or the zap is in error. Ensure that the zap has not been previously applied before calling SAS Institute Technical Support.

LSCZ246

Invalid zap number range. Range is 0-9999 - <zap-name>.

Level: ERROR

The last four bytes of the zap name was not in the range of 0-9999. The zap name must contain a number from 0-9999 in the last four bytes in order for APPLYZAP to process the zap.

LSCZ247

Library specified is not of the correct format - <library-name>.

Level: WARNING

The specified library was found to be in the wrong format. For example, APPLYZAP thought that it should have been an object module library, but found that it was instead a load module library. APPLYZAP will make the internal adjustments and continue processing the zap.

LSCZ248

Invalid command. Command is ignored - <command>.

Level: WARNING

The specified command is not valid. It will be ignored and processing will continue.

LSCZ249

Unable to open APPLYZAP input file - <DDname | filename>.

Level: WARNING

The specified input file in the INPUT option or the APPLY command could not be opened. Check to ensure that the file exists and that the filename was spelled correctly. Under MVS, ensure that the SYSLIB DD defines the library in which the file resides. Under VM, if the file is not in a MACLIB, ensure that it has a filetype of ZAP and is on an accessed disk; if the file is in a MACLIB, ensure that the SYSZLIB FILEDEF defines the MACLIB where the file resides.

LSCZ250

APPLYZAP command processed - <command>.

Level: NOTE

The APPLYZAP command <command> was processed.

LSCZ251

APPLYZAP command out of sequence.

Level: ERROR

An APPLYZAP command was found within a zap segment. APPLYZAP commands (APPLY, REPORT, or SYSTEM) can only reside outside of a zap segment. Because a zap segment is bracketed by a NAME statement and an IDRDATA or LOG statement, APPLYZAP commands cannot appear between a NAME statement and an IDRDATA or LOG statement. The command is ignored and processing of the zap continues.

LSCZ252

Zap library file does not exist - SYSZLIB.

Level: ERROR

The INPUT option or APPLY command was processed and the DD for the zap library, SYSZLIB, was not defined. Under VM, this message also indicates that a file with a filetype of <zap-name> and a filetype of ZAP could not be located on any accessed disk.

LSCZ253

Input obtained from - <SYSZLIB(zap-name) | zap-name ZAP>.

Level: NOTE

The zap input was obtained from SYSZLIB(zap-name) or on VM from zap-name ZAP.

LSCZ254

*****No zap obtained and eof not found*** - terminating.**

Level: ERROR

APPLYZAP has failed on an internal error check. Please obtain the listings from SYSZOUT, SYSPRINT, SYSTEM, and the MVS JES log or VM console log before contacting SAS Institute Technical Support.

LSCZ255

Error returned from system routine - <command> RC=nn.

Level: ERROR

The return code nn was returned from the system function when trying to execute the command. Check the particular documentation for information on the specified commands.

LSCZ256

Zap cannot be applied - Original zap not applied <zap-name>.

Level: WARNING

When attempting to apply a removal zap, it was discovered that the original (bad zap) had never been applied. Processing of the zap ends because there is no need to remove a zap that has never been applied.

LSCZ258

Applying current zap(/segments).

Level: NOTE

An APPLYZAP command (for example, APPLY, REPORT, or SYSTEM) has been found outside of a zap segment; however, it cannot be determined if the command is within the current zap (between 2 zap segments) or if the command is beginning a new zap. APPLYZAP commands should not be intermixed with zap statements. This message is issued to inform that the zap may be incorrectly split.

LSCZ259

Invalid option specified - <option>.

Level: WARNING

The specified option is not a valid APPLYZAP or utility option.

LSCZ260

Required library on the NAME card does not exist - <libname > rc = <rc>.

Level: ERROR

The *libname* specified in the NAME statement could not be found. On MVS, check to make sure a DD card for *<libname >* was given in the JCL. On CMS, if a *<.ext >* was supplied in the *libname*, ensure that the corresponding file is on a write accessible disk. On CMS, if the *<.ext >* was NOT given on the *libname*, ensure that a FILEDEF for *<libname>* is specified and points to a file on a write accessible disk.

LSCZ261

AR370 Library Manager issued diagnostics - <libname > rc = <rc>.

Level: ERROR || WARNING

Refer to the SYSPRINT and SYSZOUT files for the complete message from AR370 and take the appropriate actions. Also, consult the SAS Technical Report C-112,

"Changes and Enhancements to the SAS/CCompiler, Library, and Debugger, Release 5.50".

References

IBM (1987), *IBM MVS/XA System Programming Library: Service Aids (GC28-1159)*

SAS Institute Inc. (1989), *SAS/C Compiler and Library User's Guide, Third Edition, Cary, NC: SAS Institute Inc.*

Using OBJZAP on MVS

<i>Introduction</i>	63
<i>Control Statements</i>	63
<i>Organization</i>	63
<i>Statement Descriptions</i>	64
<i>Detailed Syntax Rules</i>	67
<i>JCL Execution</i>	67
<i>Interactive Execution</i>	68

Introduction

OBJZAP is a utility that can be run under MVS to make changes to object modules (because several parts of the products are supplied in object form). OBJZAP operates on object modules that are members of partitioned data sets, allowing users to

- examine data and instructions
- change data and instructions
- dump any control section (CSECT) of the object deck.

Although OBJZAP was designed to be similar to the IBM utility program AMASPZAP (also known as superzap), several important differences exist, notably in the algorithm used by the CHECKSUM statement. OBJZAP uses several control statements that specify the operations to be performed. This section includes a brief synopsis of the control statements and detailed syntax rules, followed by an example of an MVS job stream to run OBJZAP in batch.

Control Statements

This section provides an introduction to the organization of the control statements that are used to run the OBJZAP utility. Following this introduction is a detailed description of each statement and a brief discussion of the syntax rules.

Organization

Control statements for the OBJZAP utility are listed below. (Control statements are not case sensitive.)

```

NAME member csect
VER offset expected-content
REP offset replacement-data
CHECKSUM value

```

```

ALIAS alias1,alias2, . . .
IDRDATA xxxxxxxxxxxx
DUMP member csect

```

Each set of control statements begins with a NAME statement. The VER, REP, ALIAS, IDRDATA, and DUMP statements following the NAME statement apply to the partitioned data set (PDS) member and the CSECT that the NAME statement identifies.

The VER, REP, ALIAS, IDRDATA, and DUMP statements can appear in any order, but it is better to code all VER statements before the first REP statement to ensure that the data are verified before anything is replaced. The DUMP statement must follow the NAME statement; the CHECKSUM statement can appear anywhere in the sequence of statements. When a new NAME statement occurs, it defines a new CSECT (possibly in a new member) as the object of succeeding VER, REP, ALIAS, IDRDATA, and DUMP statements.

Statement Descriptions

□ NAME *member csect*

NAME gives the identity of the object deck *member* containing the *csect* (control section) that all succeeding control statements operate on. There is no restriction on the number of NAME statements that can appear.

□ VER *offset expected-content*

VER compares the contents of a location in the CSECT (*offset*) with the expected content (*expected-content*) supplied by the user. If the two fields do not compare as equal, the VER operation fails and a formatted dump is provided for the *csect*. No further REP operations are performed until the next NAME statement occurs. *offset* contains the hexadecimal displacement of the data in *csect*. The VER operation fails if *offset* is outside the boundaries for the CSECT specified by the NAME statement (*offset* can be an even or odd number of digits).

expected-content

contains the hexadecimal representation of the data expected at the offset in *csect*. The data must be expressed as an even number of two hexadecimal values, for example,

```
4741D175 (or, with commas: 4741,D175).
```

If commas are used to separate data, the number of digits between them must also be even. Blanks cannot be used to separate digits. A blank ends *expected-content*; any data following a blank are treated as comments and ignored. For data that will not fit in one 80-byte VER statement, a second statement must be used.

□ REP *offset replacement-data*

REP changes the data in a CSECT defined in the NAME statement. It replaces the data specified at *offset* with the data specified in *replacement-data*. The REP operation fails if *offset* is outside the boundaries for the CSECT specified in the NAME statement. The formats of the arguments to REP follow the same rules as the formats of the VER arguments.

The VER operation should always be performed to determine what will be changed with the REP function. If more than one VER and REP operation is to be performed on a CSECT, statements should be ordered so that all VER statements appear before all REP statements. The reason for this order is to ensure that no REP statement is performed if any VER operation fails.

When REP is successful, the old data are printed out, and the IDR information in the object deck is automatically updated. (See the IDRDATA discussion.)

□ **CHECKSUM <value>**

The CHECKSUM statement performs one of two tasks, depending on whether the optional *value* argument is used. The *value* argument must be eight hexadecimal digits and cannot contain commas.

If the *value* argument is present, the statement compares the number specified in *value* and the accumulated CHECKSUM. The checking is done when OBJZAP reads the control statements. If the accumulated CHECKSUM and the number specified by *value* are not equal, no processing is done.

If the *value* argument is not present, the accumulated CHECKSUM is printed in hexadecimal.

The accumulated CHECKSUM starts at zero and is reset to zero by each CHECKSUM statement. Only the offsets and data from REP statements are used in accumulating the CHECKSUM.

The CHECKSUM statement guards effectively against typographical errors in making a change. All fixes to object modules supplied by SAS Institute contain CHECKSUM statements, and you should not remove them.

You can use a comment in the CHECKSUM statement if the CHECKSUM statement contains a *value* argument. (See the discussion of comments later in this chapter.)

□ **ALIAS *alias1*<,*alias2*, . . .>**

The ALIAS statement establishes aliases for the PDS member being zapped. Each ALIAS statement can have one or more *alias* values separated by commas; no blanks can occur between aliases in an ALIAS statement. Up to sixteen unique aliases are allowed for each member.

The ALIAS statement uses the member name from the latest NAME statement. ALIAS statements can occur anywhere after the NAME statement; however, ALIAS information is updated only if there has been a successful REP operation (that is, if IDR data are updated).

Note: The ALIAS statement is needed because OBJZAP does not update in place. Therefore, aliases must be explicitly re-established; the ALIAS statement does this. Zaps to the resident portion of the run-time library from SAS Institute contain the names of any necessary aliases in an ALIAS statement as part of the zap.

□ **IDRDATA *xxxxxxxxxx***

The IDRDATA control statement is executed only if at least one REP operation is executed. IDRDATA puts a maximum of 10 bytes of user data into bytes 1 to 10 at the location of the second IDR item (on the END card of the deck that contains the CSECT). *xxxxxxxxxx* is the ten bytes of data, expressed without embedded blanks. Blanks are added at the right if less than 10 bytes are specified. IDRDATA is useful for tracing what zaps have been applied.

Note that when a REP operation is performed, the following occurs:

- 1 no IDRDATA operation is specified, *UNKNOWN* is inserted in bytes 1– 9 of the second IDR field.
- 2 If an error occurs later, the string ER is inserted in bytes 11– 12 on the END card. Otherwise, bytes 11– 12 are blanked out.
- 3 The Julian date (*yyddd*) is inserted in bytes 15– 19 on the END & card.

If there is more than one CSECT in the deck on which a REP is performed, the IDRDATA from the last CSECT is used. If there was no previous IDR statement, *UNKNOWN* is used. It is customary for zaps supplied by SAS Institute to contain IDRDATA statements with the release and zap numbers as identification.

□ **DUMP** <*member*> <*csect*>

DUMP or DUMPT dumps the CSECT identified in the NAME statement. The *csect* and *member* arguments are optional, and if specified, they must be the same as in the NAME statement. The output of the DUMP command is in hexadecimal format. When this command is used, the IDR data from the most recent zap (successful or unsuccessful) are printed, in addition to the contents of the CSECT. Note that a NAME statement must always come before the DUMP statement.

□ **GO**

GO delimits a set of control statements. It is primarily intended for interactive execution of OBJZAP under TSO.

The GO statement instructs OBJZAP to perform the preceding control statements. OBJZAP does syntax checking and executes the statements if it finds no syntax errors. You can then enter a new set of OBJZAP control statements. (The new set must begin with a NAME statement.)

□ * **Comments**

Comment statements: You can use comments in the OBJZAP command stream. Comment statements must be in the form **comment*. (The number of comments is not limited.) OBJZAP writes the comment statements to the standard output data set (**stdout**).

Comments included on control statements: You can also include comments on control statements, other than the DUMP or DUMPT statements. In a control statement, place the comment after the last argument that the statement requires. Precede the comment with a blank. You do not need an asterisk (*) to indicate a comment on a control statement. If no arguments are present, you cannot use a comment on the control statement.

Detailed Syntax Rules

As described earlier in this chapter, statement syntax is similar to that for the IBM program AMASPZAP. The detailed syntax rules follow:

- An OBJZAP operation name must be specified before any arguments in a statement.
- The statement can be entered starting in any column. Control statements can be up to 80 bytes long. (Information beyond column 72 is ignored.)
- Several blanks can separate the OBJZAP operation name and its first argument, but there must be at least one blank. Similarly, one or more blanks must separate arguments in the statement.
- Commas can be used in data fields other than offsets and the CHECKSUM *value*. Blanks are not allowed inside data fields. A blank terminates a data field.
- Values in the *expected-content* field (VER statement) and the *replacement-data* field (REP statement) must be expressed as an even number of hexadecimal digits.
- Comment statements are specified by an asterisk. Comments can also be used in control statements other than the DUMP statement. In a control statement, a space separates the comment from the last argument that the statement takes. No asterisk (*) is needed for comments on control statements.
- Control statements are not case sensitive.

JCL Execution

The following DD statements are required to execute OBJZAP:

SYSIN

contains the OBJZAP control statements.

SYSPRINT

identifies the standard output message data set (sequential).

SYSLIB

identifies the partitioned data set containing the member(s) to be zapped.

STEPLIB

compiler library and transient library.

The following statement is optional:

SYSTEM the data set to be used for diagnostic messages.

Example B.1 shows a sample MVS JCL routine to run OBJZAP.

Example Code A2.1 Sample JCL for OBJZAP

```
//jobname JOB jobcard information
//OBJZAP EXEC PGM=OBJZAP
//STEPLIB DD DSN=sasc.LOAD,DISP=SHR
//          DD DSN=sasc.LINKLIB,DISP=SHR
//SYSIN DD *
```

```
NAME XXX BUILD@  
VER 002 C5E2C3  
REP 002 C6E3C4  
IDRDATA ABCDEF1237  
* A NEW CSECT IN A DIFFERENT PDS MEMBER.  
NAME ZZZ MAIN@  
VER 35 969540F1  
VER 6D A3009096  
IDRDATA XXXXEF1237  
//SYSLIB DD DSN=sasc.object.library,DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SYSTEM DD SYSOUT=*  
//
```

In batch mode, OBJZAP reads all control statements before processing them.

Interactive Execution

You can run OBJZAP interactively under TSO by entering OBJZAP control statements from the terminal, followed by a GO statement. The GO statement executes all control statements you enter before the GO statement. Use the string EOF (in uppercase) to signal end-of-input when you are finished with OBJZAP.

Index

A

ALIAS statement 42
 AMASPZAP utility
 definition 11
 zapping load module libraries 13
 APPLY option 14
 APPLYZAP procedure 16
 APPLYZAP utility
 definition 11
 error messages 34
 filename extensions 14
 input/output files 14
 JCL sample 16
 listing applied zaps 18
 MVS DDnames 15
 MVS options 13
 MVS zaps 13
 return codes 33
 summary table sample 18
 technical description 13
 VM options 17
 VM zaps 17
 with APPLYZAP procedure 16
 zap processing 15
 zapping multiple MVS libraries 16
 zapping multiple VM libraries 17
 AR370 archived members 13

C

CHECK option 13
 CHECKSUM statement 13, 42
 CHECKSUMs, validating 13
 code generation problems 4
 comments (*) statement 43
 compiler problems 4

D

dash (-), in APPLYZAP utility 13
 diagnostic information

collecting 4
 echoing 13
 reporting 6
 downloading SAS/C files 26
 DUMP statement 42

E

e-mail
 automatic updates from 29
 reporting problems 29
 ECHO option 13
 Electronic Mail Interface to Technical Support (EMITS)
 29
 electronic support services
 See also technical support
 e-mail, automatic updates from 29
 e-mail, reporting problems 29
 Electronic Mail Interface to Technical Support
 (EMITS) 29
 FTP 28
 problem reports, submitting 25
 problem reports, updating 26
 SAS Web site 23
 SAS/C files, downloading 26
 Technical Support database 24
 Technical Support News Listing (TSNEWS-L) 29
 EMITS (Electronic Mail Interface to Technical Support)
 29
 error messages 34
 ERROR option 14
 error tolerance, enabling 14

F

FILEDEF option 14
 filename extensions 14
 FTP 28

G

GO statement 43

I

IDRDATA statement 13, 42

70 Index

input file
 overriding default 14
 specifying 14
INPUT option 14
input/output files 14
Installation Coordinator 3

J

JCL 16, 43

L

libraries, zapping
 load module libraries 13
 multiple MVS libraries 16
 multiple VM libraries 17
 object module libraries 13
link problems 5
linkage editor default, overriding 14
LINKNAME option 14
LIST option 14
listing zaps 14, 18
load module libraries, zapping 13
LOADLIBs 13
LOG statement 13

M

maintenance categories 9
maintenance updates
 automatic application 18
 obtaining 6
MODULES 13
MVS zaps 13
 APPLYZAP utility 13
 automatic maintenance updates 19
 general information 13
 OBJZAP utility 41
 required DDnames 15
 zapping multiple libraries 16

N

NAME statement 13, 41
NOT sign (!), in APPLYZAP utility 13

O

object module libraries, zapping 13
OBJZAP utility
 ALIAS statement 42
 CHECKSUM statement 42
 definition 11
 DUMP statement 42
 GO statement 43
 IDRDATA statement 42
 interactive execution 43
 JCL execution 43
 MVS zaps 41
 NAME statement 41
 REP statement 42
 syntax 43
 VER statement 41
 zapping object module libraries 13
 * (comments) statement 43
output
 directing to a terminal 14
 uppercasing 14
OVERRIDE option 14

P

PRECK option 14
problem reports
 submitting 25
 updating 26
product number 3
product support 6

R

RECOMMENDED maintenance 9
REMOVE maintenance 9
REP statement 42
REPORT option 14
REPORT ZAP option 14
REQUIRED maintenance 9
return codes 33

S

sample library 21
SAS/C files, downloading 26

SAS/C Installation Coordinator 3
 SAS/C sample library 21
 SAS/C Software Consultant 3
 SAS Communications magazine 7
 SAS Users Group International (SUGI) 7
 SAS Web site 23
 site number 4
 Software Consultant 3
 SPECIAL CONSIDERATION maintenance 9
 SUGI (SAS Users Group International) 7
 summary table 18
 Support Representative 3
 SYSIN files 15
 SYSPRINT files 15
 system commands, passing 14
 SYSTEM option 14
 SYSZIN files 14
 SYSZLIB files 14
 SYSZLIN files 15
 SYSZOUT files 15
 SYSZRPT files 15

T

technical support
 See also electronic support services
 e-mail address 29
 reasons for calling 3
 required information 4
 telephone numbers 7
 Technical Support database 24
 Technical Support News Listing (TSNEWS-L) 29
 TERM option 14
 terminal, directing output to 14
 TEXT files 13
 TXTLIBs 13
 tracebacks, reading 5
 TSNEWS-L (Technical Support News Listing) 29

U

updates, maintenance
 automatic application 18
 obtaining 6
 UPPER option 14

uppercasing output 14
 usage notes 6
 Usage Notes files 9
 user feedback 7

V

VER statement 41
 version number 4
 VM zaps 17
 APPLYZAP utility 17
 automatic maintenance updates 20
 general information 13
 ZAP utility 13
 zapping multiple libraries 17
 ZAPTEXT utility 13

W

Web site 23

Z

zap information reports 14
 zap maintenance categories 9
 zap processing 15
 zap utilities 14
 See also AMASPZAP utility
 See also APPLYZAP utility
 See also OBJZAP utility
 See also ZAP utility
 See also ZAPTEXT utility
 ZAP utility
 definition 11
 VM zaps 13
 ZAPNAME option 14
 ZAPPED files 15
 zaps
 applying 11
 AR370 archived members 13
 directing output to a terminal 14
 duplicate application 14
 input file, specifying 14
 listing 14, 18
 LOADLIBs 13
 MODULES 13

72 Index

- prerequisites, checking for 14
- system commands, passing 14
- TEXT files 13
- TXTLIBs 13
- uppercasing output 14
- zaps, MVS operating environment
 - See MVS zaps
- zaps, VM operating environment
 - See VM zaps
- zaps files 9
- ZAPTEXT utility
 - definition 11
 - VM zaps 13

Special Characters

- ! (NOT sign), in APPLYZAP utility 13
- * (comments) statement 43
- (dash), in APPLYZAP utility 43

Your Turn

If you have comments or suggestions about *A Guide for the SAS/C Compiler Consultant, Release 6.50*, please send them to us on a photocopy of this page or send us electronic mail.

Send comments about this book to

SAS Institute Inc.
Publications Division
SAS Campus Drive
Cary, NC 27513
email: **yourturn@unx.sas.com**

Send suggestions about the software to

SAS Institute Inc.
Technical Support Division
SAS Campus Drive
Cary, NC 27513
email: **suggest@unx.sas.com**