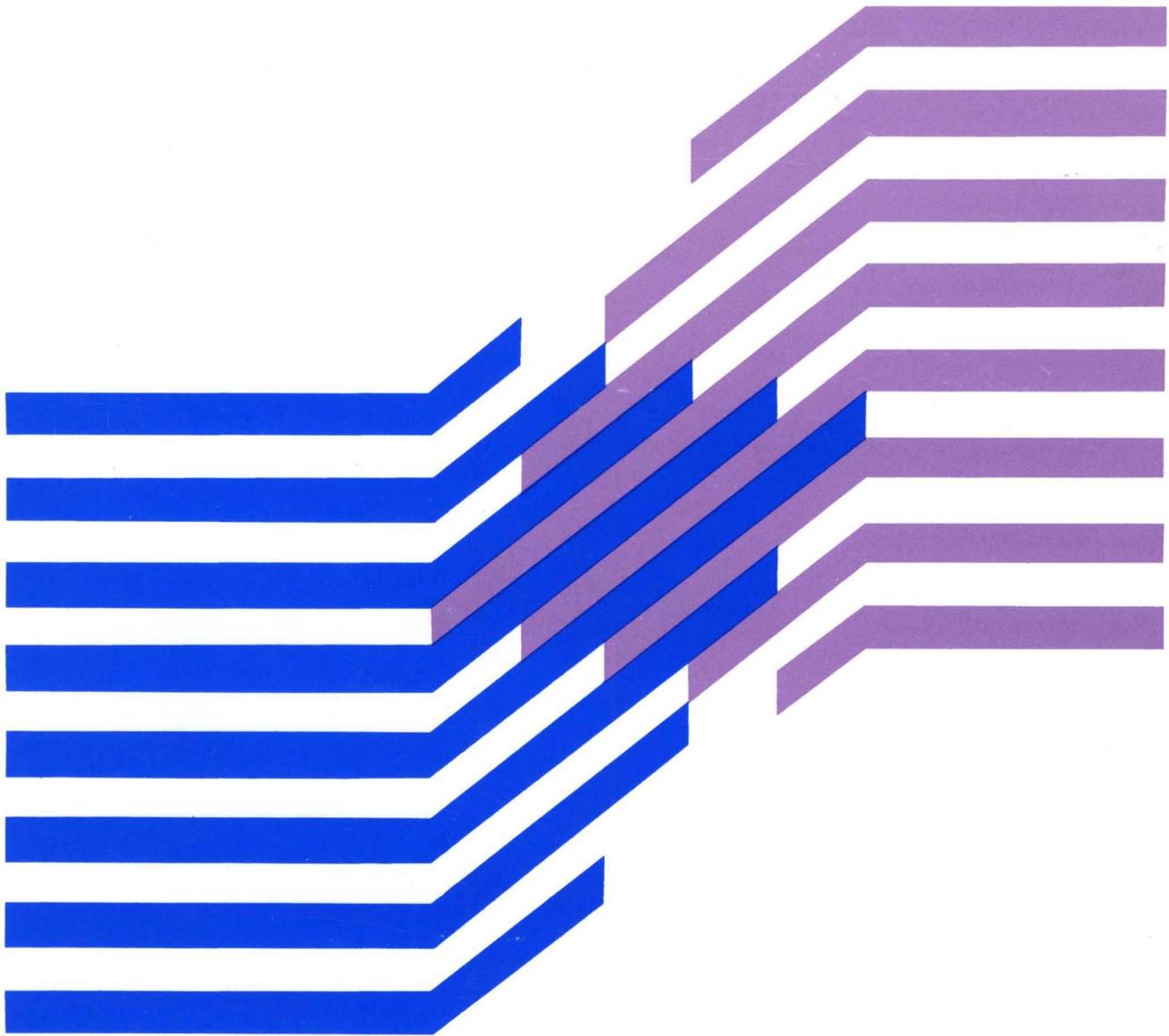




MVS/ESA  
Component Diagnosis:  
Callable Service Requests

MVS System Product:  
JES2 Version 3  
JES3 Version 3





---

MVS/ESA

LY28-1024-1

Component Diagnosis:  
Callable Service Requests

MVS System Product:

JES2 Version 3

JES3 Version 3

---

**Production of This Book**

This book was prepared and formatted using the IBM BookMaster document markup language.

**Second Edition (December, 1989)**

This is a major revision of, and obsoletes, LY28-1024-0. See the Summary of Changes following the Contents for a summary of the changes made to this manual. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

This edition applies to Version 3 Release 1.3 of MVS/System Product 5685-001 or 5685-002 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 Bibliography*, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products or services do not imply that IBM intends to make these available in all countries in which IBM operates. References to IBM products in this document do not imply that functionally equivalent products may be used. The security certification of the trusted computing base that includes the products discussed herein covers certain IBM products. Please contact the manufacturer of any product you may consider to be functionally equivalent for information on that product's security classification. This statement does not expressly or implicitly waive any intellectual property right IBM may hold in any product mentioned herein.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Information Development, Department D58, Building 921-2, PO Box 950, Poughkeepsie, NY 12602. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

**PROGRAMMING INTERFACES**

**This book consists entirely of diagnostic information.  
Such information should never be used as programming  
interface information.**



---

## About This Book

This book will help you diagnose problems in the callable service requests component. Using the book, you will be able to:

- Follow diagnostic procedures for each type of problem in the component
- Collect and analyze data needed to diagnose the problem
- Develop a search argument to be used for searching problem reporting data bases
- Know what problem data is needed before reporting the problem to IBM

---

## Trademarks

The following are trademarks of International Business Machines Corporation.

- MVS/ESA™
- Hiperspace™

The following are registered trademarks of International Business Machines Corporation.

- BookMaster®

---

## Who Should Use This Book

This book is for anyone who diagnoses problems that appear to be caused by callable service requests.

The level of detail at which this book is written assumes that the reader:

- Understands basic system concepts and the use of system services
- Codes in assembler language, and reads assembler and linkage editor output
- Codes JCL statements for batch jobs and cataloged procedures
- Understands the commonly used diagnostic tasks and aids, such as message logs, dumps, and the interactive problem control system (IPCS)
- Understands the externals for callable service requests

---

## How to Use this Book

Before using this book, collect the following problem data. If you do not have this data, see the *Basics of Problem Determination* book and perform its procedures.

- The problem type, such as an abend
- The product name
- The component name: callable service requests

Use this book to diagnose problems only in callable service requests. If the component name is not callable service requests, return to the *Basics of Problem Determination* book to identify the component. Use the component diagnosis book for the identified component.

Use this book to diagnose problems in the component as follows:

1. Identify the problem type.
2. Collect problem data.
3. Analyze the problem data to develop symptoms and to identify the failing code.
4. Fix the problem, if it is in installation-provided code.
5. Develop search arguments, search problem reporting data bases, and request the problem fix, if the problem has been reported before. If not, continue diagnosis.
6. Collect additional problem data.
7. Analyze the problem data to isolate the problem.
8. Report the problem to IBM, if assistance is needed or if the problem is new.

---

## Where to Find More Information

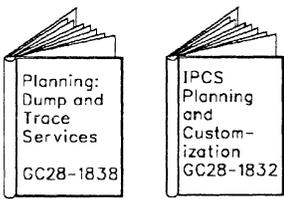
The following table lists books that contain information related to the information contained in this book.

When this book references information in other books, the shortened version of the book title is used. The following table shows the shortened titles, complete titles, and order numbers of the books that you might need while you are using this book.

Short Title Used in This Book	Title	Order Number
Application Development Guide	MVS/ESA Application Development Guide	GC28-1821
Application Development Macro Reference	MVS/ESA Application Development Macro Reference	GC28-1822
Basics of Problem Determination	MVS/ESA Basics of Problem Determination	GC28-1839
Component Diagnosis: Data-in-Virtual	MVS/ESA Component Diagnosis: Data-in-Virtual	LY28-1456
EREP User's Guide	Environmental Record Editing and Printing Program User's Guide and Reference	GC28-1378
IPCS Command Reference	MVS/ESA Interactive Problem Control System (IPCS) Command Reference	GC28-1834
IPCS User's Guide	MVS/ESA Interactive Problem Control System (IPCS) User's Guide	GC28-1833
SPL: Application Development - Extended Addressability	MVS/ESA System Programming Library: Application Development - Extended Addressability	GC28-1854
SPL: Application Development Guide	MVS/ESA System Programming Library: Application Development Guide	GC28-1852
SPL: Application Development Macro Reference	MVS/ESA System Programming Library: Application Development Macro Reference	GC28-1857
System Codes	MVS/ESA Message Library: System Codes	GC28-1815
System Commands	MVS/ESA Operations: System Commands	GC28-1826
System Messages	MVS/ESA Message Library: System Messages, Volumes 1 - 2	GC28-1812 and GC28-1813
Using Dumps and Traces	MVS/ESA Diagnosis: Using Dumps and Traces	LY28-1843

## Diagnosis Library

**PLANNING:**

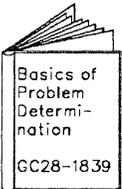


Planning: Dump and Trace Services  
GC28-1838

IPCS Planning and Customization  
GC28-1832

**DIAGNOSIS:**

For Diagnostic Procedures



Basics of Problem Determination  
GC28-1839

For Data Collection



Diagnosis: Using Dumps and Traces  
LY28-1843

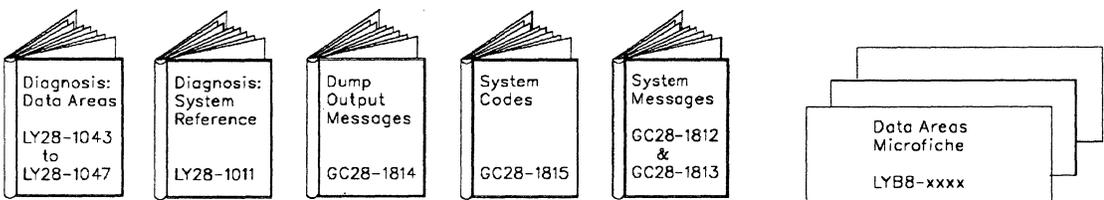
Service Aids  
GC28-1844

SYS1.LOGREC Error Recording  
GC28-1845

IPCS User's Guide  
GC28-1833

IPCS Command Reference  
GC28-1834

For Data Interpretation



Diagnosis: Data Areas  
LY28-1043 to LY28-1047

Diagnosis: System Reference  
LY28-1011

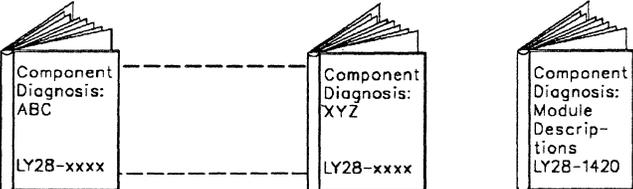
Dump Output Messages  
GC28-1814

System Codes  
GC28-1815

System Messages  
GC28-1812 & GC28-1813

Data Areas Microfiche  
LY88-xxxx

For Component-Specific Information  
(To identify the component, see Basics of Problem Determination)



Component Diagnosis: ABC  
LY28-xxxx

Component Diagnosis: XYZ  
LY28-xxxx

Component Diagnosis: Module Descriptions  
LY28-1420



---

## Contents

<b>Chapter 1. Diagnosing Problems in Callable Service Requests</b> . . . . .	1-1
Identifying Problems . . . . .	1-1
Diagnosing an Abend Unexpected by Cell Pool Services . . . . .	1-5
Diagnosing an Abend Unexpected by Data Windowing Services . . . . .	1-8
Diagnosing a Return Code Due to a Problem in Cell Pool Services . . . . .	1-11
Diagnosing a Return Code Due to a Problem in Data Windowing Services . . . . .	1-13
<b>Chapter 2. Collecting and Analyzing Problem Data for Callable Service Requests</b> . . . . .	2-1
Using the IPCS CBSTAT and CBFORMAT Subcommands to Obtain Cell Pool Services Data . . . . .	2-1
<b>Chapter 3. Developing a Search Argument for Callable Service Requests</b> . . . . .	3-1
<b>Chapter 4. Reporting a Problem to IBM</b> . . . . .	4-1
<b>Chapter 5. Callable Service Requests Functions</b> . . . . .	5-1
Data Windowing Services . . . . .	5-1
Data Windowing Services Functions . . . . .	5-1
Cell Pool Services . . . . .	5-2
Cell Pool Services Functions . . . . .	5-2
Cell Pool Exits for IPCS . . . . .	5-2
Callable Service Requests Modules . . . . .	5-3
<b>Index</b> . . . . .	X-1



---

## Summary of Changes

|  
|  
|

**Summary of Changes  
for LY28-1024-1  
MVS/System Product Version 3 Release 1.3**

|  
|

The following summarizes the changes to the book.

|  
|

***New Information:*** Information was added to support cell pool services.

|  
|  
|

***Changed Information:*** Chapters 2 through 4 are renumbered to 3 through 5 and a new Chapter 2, “Collecting and Analyzing Problem Data for Callable Service Requests” is added.

|

**Summary of Changes  
for LY28-1024-0  
MVS/System Product Version 3 Release 1**

This book contains diagnostic information for callable service requests. This component provides callable interfaces to support applications and system services that are not directly available from high level languages. This component is new for MVS/System Product Version 3 Release 1.



## Chapter 1. Diagnosing Problems in Callable Service Requests

### Prerequisites

Before using this book, collect the following problem data. If you do not have this data, see the *Basics of Problem Determination* book and perform its procedures.

- The problem type, such as an abend
- The product name
- The component name: callable service requests

Use this book to diagnose problems only in callable service requests. If the component name is not callable service requests, return to the *Basics of Problem Determination* book to identify the component. Use the component diagnosis book for the identified component.

## Identifying Problems

For a problem with these symptoms:	See the following:
<p><b>Abend X'019'</b>: Callable service requests abnormally terminates a caller with a system completion code of X'019'. This indicates a user problem. An accompanying reason code indicates the problem.</p> <p><b>Note:</b> Abend X'019' with reason code 0 occurs when you try to run a callable service on a lower release level system than the one it was compiled on.</p>	<p><i>Application Development Guide</i> for callable service requests and the reason codes</p>
<p><b>Abend X'08B'</b>: A module with the prefix CSR abnormally terminates with a system completion code of X'08B'.</p>	<p><i>Component Diagnosis: Data-in-Virtual</i></p>
<p><b>Abend unexpected by cell pool services:</b> A module with the prefix CSRPC abnormally terminates with any code. This is an unexpected problem.</p>	<p>"Diagnosing an Abend Unexpected by Cell Pool Services" on page 1-5</p>
<p><b>Abend unexpected by data windowing services:</b> A module with the prefix CSR abnormally terminates with a code other than X'019' or X'08B'. This is an unexpected problem.</p>	<p>"Diagnosing an Abend Unexpected by Data Windowing Services" on page 1-8</p>
<p><b>Return code from the cell pool services part of callable service requests:</b> The following return codes from the cell pool services part of callable service requests are for information only and do not indicate a problem. The return codes are returned to the program issuing the call and appear in the caller's parameter list and register 15.</p> <p><b>Return Code</b> hex(dec)</p> <p>4 (4) 8 (8) C (12)</p>	<p><i>SPL: Application Development Guide</i> for an explanation of the return codes</p>

For a problem with these symptoms:	See the following:								
<p><b>Return code due to a user problem in the cell pool services part of callable service requests:</b> The following return codes indicate a user problem in the cell pool services part of callable service requests. The return codes are returned to the program issuing the call and appear in the caller's parameter list and register 15.</p> <p><b>Return Code</b> hex(dec)</p> <p>18 (24) 1C (28) 28 (40) 2C (44) 30 (48) 34 (52) 38 (56) 44 (68) 48 (72) 4C (76) 50 (80) 54 (84) 58 (88)</p>	<p><i>Application Development Macro Reference</i> for the description of the cell pool service being executed</p>								
<p><b>Return code due to problems in the cell pool services part of callable service requests:</b> The following return codes indicate a problem in the cell pool services part of callable service requests. The return codes are returned to the program issuing the call and appear in the caller's parameter list and register 15.</p> <p><b>Return Code</b> hex(dec)</p> <p>64 (100) 68 (104) 6C (108) 70 (112) 74 (116)</p>	<p>"Diagnosing a Return Code Due to a Problem in Cell Pool Services" on page 1-11</p>								
<p><b>Return and reason codes due to a problem in the data windowing services part of callable service requests:</b> The following return and reason code combinations indicate a problem in the data windowing services part of callable service requests. The return codes are returned to the program issuing the call and appear in the caller's parameter list and register 15. The reason codes are returned to the program issuing the call and appear in the caller's parameter list and register 0.</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Return Code</th> <th style="text-align: left;">Reason Codes</th> </tr> <tr> <th style="text-align: left;">hex(dec)</th> <th style="text-align: left;">(hex)</th> </tr> </thead> <tbody> <tr> <td>4 (4)</td> <td>00000125</td> </tr> <tr> <td>8 (8)</td> <td>00000118, 00000119, 00000152</td> </tr> </tbody> </table>	Return Code	Reason Codes	hex(dec)	(hex)	4 (4)	00000125	8 (8)	00000118, 00000119, 00000152	<p>"Diagnosing a Return Code Due to a Problem in Data Windowing Services" on page 1-13</p>
Return Code	Reason Codes								
hex(dec)	(hex)								
4 (4)	00000125								
8 (8)	00000118, 00000119, 00000152								

For a problem with these symptoms:	See the following:						
<p><b>Return and reason code due to a user problem in the data windowing services part of callable service requests:</b> The following code indicates a user problem, such as a problem in the program issuing the CALL statement. The return code is returned to the program issuing the call and appear in the caller's parameter list and register 15. The reason code is returned to the program issuing the call and appear in the caller's parameter list and register 0.</p> <table border="0"> <tr> <td><b>Return Code</b></td> <td><b>Reason Code</b></td> </tr> <tr> <td>hex(dec)</td> <td>(hex)</td> </tr> <tr> <td>8 (8)</td> <td>00000143</td> </tr> </table>	<b>Return Code</b>	<b>Reason Code</b>	hex(dec)	(hex)	8 (8)	00000143	<p><i>Application Development Macro Reference</i> for the description of the callable service routine being executed</p>
<b>Return Code</b>	<b>Reason Code</b>						
hex(dec)	(hex)						
8 (8)	00000143						
<p><b>Return code from the data windowing services part of callable requests due to a problem in the data-in-virtual component:</b> The following return and reason code combinations from the data windowing services part of callable service requests indicate a problem in the data-in-virtual component. The return codes are returned to the program issuing the call and appear in the caller's parameter list and register 15. The reason code is returned to the program issuing the call and appear in the caller's parameter list and register 0.</p> <table border="0"> <tr> <td><b>Return Code</b></td> <td><b>Reason Code</b></td> </tr> <tr> <td>hex(dec)</td> <td>(hex)</td> </tr> <tr> <td>4 (4) C (12)</td> <td>hhhhhhhh</td> </tr> </table>	<b>Return Code</b>	<b>Reason Code</b>	hex(dec)	(hex)	4 (4) C (12)	hhhhhhhh	<p><i>Component Diagnosis: Data-in-Virtual</i> for an explanation of the return and reason codes</p>
<b>Return Code</b>	<b>Reason Code</b>						
hex(dec)	(hex)						
4 (4) C (12)	hhhhhhhh						
<p><b>Return code following an SVC 99:</b> The following return and reason code from the data windowing services part of callable service requests after an SVC 99 indicates a problem in the allocation/unallocation component or in the Data Facility Product (DFP), including the storage management subsystem (SMS). The return code is returned to the program issuing the call and appears in the caller's parameter list and register 15. The reason code is returned to the program issuing the call and appears in the caller's parameter list and register 0.</p> <table border="0"> <tr> <td><b>Return Code</b></td> <td><b>Reason Code</b></td> </tr> <tr> <td>hex(dec)</td> <td>(hex)</td> </tr> <tr> <td>10 (16)</td> <td>hhhhhhhh</td> </tr> </table>	<b>Return Code</b>	<b>Reason Code</b>	hex(dec)	(hex)	10 (16)	hhhhhhhh	<p><i>SPL: Application Development Guide</i> for an explanation of the return and reason codes</p>
<b>Return Code</b>	<b>Reason Code</b>						
hex(dec)	(hex)						
10 (16)	hhhhhhhh						

## Diagnosing an Abend Unexpected by Cell Pool Services

A module with the prefix CSRCP abnormally terminates with any code. This is an unexpected problem.

Callable service requests has no error recovery. Any dumps will appear to be for a problem in the caller or data-in-virtual.

If the instruction at time of error is in callable service requests, check for a problem in the caller's parameter list.

Another possible cause of the problem is that the application issuing the CALL for a callable service did not check the return code after the CALL; then the application issued another CALL to another callable service using incorrect output from the first CALL.

Diagnostic procedure	References
<p>1. <b>Look at the explanation for the abend code and any accompanying reason code.</b> Take the recommended actions.</p>	<p><i>System Codes</i> for an explanation of the abend code</p>
<p>2. <b>Collect and analyze problem data: messages.</b> Obtain messages accompanying the abend. Look at their explanations and take any recommended actions.</p>	<p><i>System Messages</i> for explanations of the messages</p>
<p>3. <b>Collect problem data: the SYS1.LOGREC record or the SVC dump or SYSMDUMP ABEND dump for the abend.</b></p> <ul style="list-style-type: none"> <li>• Format a SYS1.LOGREC record with EREP to obtain a detail edit report.</li> <li>• Format the dump with a STATUS FAILDATA subcommand or a VERBEXIT LOGDATA subcommand.</li> </ul>	<p><i>EREP User's Guide</i> for formatting SYS1.LOGREC records</p> <p><i>IPCS Command Reference</i> for the subcommand:</p> <p style="text-align: center;">STATUS VERBEXIT LOGDATA</p>
<p>4. <b>Identify the failing input request to callable service requests.</b></p> <p>To identify the request, use the explanations for the abend code and/or the abend reason code. Also, look in OTHER SERVICEABILITY INFORMATION in the STATUS FAILDATA output, VERBEXIT LOGDATA output, or SYS1.LOGREC report.</p>	
<p>5. <b>Collect information about the cell pool</b> from the dump using the IPCS CBSTAT and CBFORMAT subcommands.</p>	<p>Chapter 2, "Collecting and Analyzing Problem Data for Callable Service Requests" on page 2-1</p>

Diagnostic procedure	References
<p>6. <b>Search a problem reporting data base</b> to determine if the problem was previously reported. Use the argument under SEARCH ARGUMENT ABSTRACT in:</p> <ul style="list-style-type: none"> <li>• SYS1.LOGREC detail edit report</li> <li>• STATUS FAILDATA output</li> <li>• VERBEXIT LOGDATA output</li> </ul> <p>If the search finds no match, remove some symptoms or change the argument to the following. Search again. Continue searching for matches by adding, removing, and changing symptoms.</p> <ul style="list-style-type: none"> <li>• Program identifier: PIDS/5752SCCSR</li> <li>• Load module name: RIDS/CSRCPccc#L</li> <li>• Module name: RIDS/CSRCPccc</li> <li>• System abend code: AB/S0hhh</li> <li>• Abend reason code: PRCS/hhhhhhhh</li> </ul> <p>If the search finds that the problem was previously reported, request the problem fix. If not, continue with the next step.</p>	<p>Chapter 3, “Developing a Search Argument for Callable Service Requests”</p>
<p>7. <b>Analyze the dump or SYS1.LOGREC report for VRA data.</b> Find the heading VARIABLE RECORDING AREA (SDWAVRA) in the STATUS FAILDATA output, VERBEXIT LOGDATA output, or SYS1.LOGREC report. Note the SDWAVRA keys, lengths, and contents.</p>	
<p>8. <b>Collect dump data.</b></p> <p>Format the dump and print it or store it in a data set. Use the following IPCS subcommands, in this order. If using IPCS interactively, respond yes to the IPCS message that asks if summary data can be used by dump access.</p> <pre> STATUS FAILDATA SYSTEM CPU REGISTERS DATA CONTENTION WORKSHEET SUMMARY TCBERROR SUMMARY FORMAT VERBEXIT LOGDATA VERBEXIT SUMDUMP VERBEXIT TRACE VERBEXIT MTRACE CBSTAT addr STRUCTURE(CSRCP00L) CBFORMAT addr STRUCTURE(CSRCP00L) VERBEXIT SYMPTOM           </pre>	<p><i>Using Dumps and Traces</i> for requesting and printing dumps</p> <p><i>IPCS Command Reference</i> for the subcommands</p> <p>Chapter 2, “Collecting and Analyzing Problem Data for Callable Service Requests”</p>

Diagnostic procedure	References
<p><b>9. Analyze the dump for the offset of the failing instruction into the module or CSECT.</b></p> <p>Obtain the address in the right half of the program status word (PSW) in the STATUS FAILDATA or VERBEXIT LOGDATA output. Subtract the instruction length from the PSW address to obtain the address of the failing instruction. If the leftmost digit in the address is greater than or equal to 8, subtract 8 from it. Then do one of the following:</p> <ul style="list-style-type: none"> <li>• If analyzing the dump interactively, use the instruction address in a WHERE subcommand to obtain the offset of the PSW address into the load module.</li> <li>• If analyzing printed output, find the instruction address: <ul style="list-style-type: none"> <li>– In dump output from the LIST or VERBEXIT SUMDUMP subcommand. Look for the CSECT name eye-catcher. IBM CSECT eye-catchers are generally followed by an assembly date and a product identifier or PTF level, such as JBB3313 or UY01234; most eye-catchers are at the beginning of a module, but some are at the end.</li> <li>– In a module listed for the LPAMAP or VERBEXIT NUCMAP subcommand.</li> </ul> </li> </ul> <p>Subtract the starting address of the CSECT from the instruction address to obtain the offset into the CSECT.</p>	<p><i>IPCS Command Reference</i> for the subcommands:</p> <p>LIST LPAMAP STATUS VERBEXIT LOGDATA VERBEXIT NUCMAP VERBEXIT SUMDUMP WHERE</p>
<p><b>10. Collect and analyze any other problem data recommended in the procedure for an abend in the <i>Basics of Problem Determination</i> book.</b></p>	
<p><b>11. Report the problem to IBM, if assistance is needed or if the problem is new. Provide the following problem data:</b></p> <ul style="list-style-type: none"> <li>• Problem type: abend</li> <li>• Search argument</li> <li>• Dump formatted by IPCS, online or printed</li> <li>• Output from CBSTAT and CBFORMAT subcommands</li> <li>• SDWAVRA keys, lengths, and contents</li> <li>• Offset of the failing instruction into the module</li> <li>• Accompanying messages: identifiers and texts</li> <li>• SYS1.LOGREC report, if used</li> <li>• Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service</li> <li>• Other problem data developed while using the <i>Basics of Problem Determination</i> book</li> </ul>	<p>Chapter 4, "Reporting a Problem to IBM"</p>

## Diagnosing an Abend Unexpected by Data Windowing Services

A module with the prefix CSR abnormally terminates with a code other than X'019' or X'08B'. This is an unexpected problem.

Callable service requests has no error recovery. Any dumps will appear to be for a problem in the caller or data-in-virtual.

If the instruction at time of error is in callable service requests, check for a problem in the caller's parameter list.

Another possible cause of the problem is that the application issuing the CALL for a callable service did not check the return code after the CALL; then the application issued another CALL to another callable service using incorrect output from the first CALL.

Diagnostic procedure	References
<p>1. Look at the explanation for the abend code and any accompanying reason code. Take the recommended actions.</p>	<p><i>System Codes</i> for an explanation of the abend code</p>
<p>2. Collect and analyze problem data: messages. Obtain messages accompanying the abend. Look at their explanations and take any recommended actions.</p>	<p><i>System Messages</i> for explanations of the messages</p>
<p>3. Collect problem data: the SYS1.LOGREC record or the SVC dump or SYSMDUMP ABEND dump for the abend.</p> <ul style="list-style-type: none"> <li>• Format a SYS1.LOGREC record with EREP to obtain a detail edit report.</li> <li>• Format the dump with a STATUS FAILDATA subcommand or a VERBEXIT LOGDATA subcommand.</li> </ul>	<p><i>EREP User's Guide</i> for formatting SYS1.LOGREC records</p> <p><i>IPCS Command Reference</i> for the subcommand:</p> <p style="text-align: center;">STATUS VERBEXIT LOGDATA</p>
<p>4. Identify the failing input request to callable service requests.</p> <p>To identify the request, use the explanations for the abend code and/or the abend reason code. Also, look in OTHER SERVICEABILITY INFORMATION in the STATUS FAILDATA output, VERBEXIT LOGDATA output, or SYS1.LOGREC report.</p>	
<p>5. Search a problem reporting data base to determine if the problem was previously reported. Use the argument under SEARCH ARGUMENT ABSTRACT in:</p> <ul style="list-style-type: none"> <li>• SYS1.LOGREC detail edit report</li> <li>• STATUS FAILDATA output</li> <li>• VERBEXIT LOGDATA output</li> </ul> <p>If the search finds no match, remove some symptoms or change the argument to the following. Search again. Continue searching for matches by adding, removing, and changing symptoms.</p> <ul style="list-style-type: none"> <li>• Program identifier: PIDS/5752SCCSR</li> <li>• Load module name: RIDS/CSRcccc#L</li> <li>• Module name: RIDS/CSRcccc</li> <li>• System abend code: AB/S0hhh</li> <li>• Abend reason code: PRCS/hhhhhhhh</li> </ul> <p>If the search finds that the problem was previously reported, request the problem fix. If not, continue with the next step.</p>	<p>Chapter 3, “Developing a Search Argument for Callable Service Requests”</p>

Diagnostic procedure	References
<p>6. <b>Analyze the dump or SYS1.LOGREC report for VRA data.</b> Find the heading VARIABLE RECORDING AREA (SDWAVRA) in the STATUS FAILDATA output, VERBEXIT LOGDATA output, or SYS1.LOGREC report. Note the SDWAVRA keys, lengths, and contents.</p>	
<p>7. <b>Collect dump data.</b></p> <p>Format the dump and print it or store it in a data set. Use the following IPCS subcommands, in this order. If using IPCS interactively, respond yes to the IPCS message that asks if summary data can be used by dump access.</p> <p style="padding-left: 40px;">STATUS FAILDATA SYSTEM CPU REGISTERS DATA CONTENTION WORKSHEET SUMMARY TCBERROR SUMMARY FORMAT VERBEXIT LOGDATA VERBEXIT SUMDUMP VERBEXIT TRACE VERBEXIT MTRACE VERBEXIT SYMPTOM</p>	<p><i>Using Dumps and Traces</i> for requesting and printing dumps</p> <p><i>IPCS Command Reference</i> for the subcommands</p> <p>Chapter 2, “Collecting and Analyzing Problem Data for Callable Service Requests”</p>
<p>8. <b>Analyze the dump for the offset of the failing instruction into the module or CSECT.</b></p> <p>Obtain the address in the right half of the program status word (PSW) in the STATUS FAILDATA or VERBEXIT LOGDATA output. Subtract the instruction length from the PSW address to obtain the address of the failing instruction. If the leftmost digit in the address is greater than or equal to 8, subtract 8 from it. Then do one of the following:</p> <ul style="list-style-type: none"> <li>• If analyzing the dump interactively, use the instruction address in a WHERE subcommand to obtain the offset of the PSW address into the load module.</li> <li>• If analyzing printed output, find the instruction address: <ul style="list-style-type: none"> <li>– In dump output from the LIST or VERBEXIT SUMDUMP subcommand. Look for the CSECT name eye-catcher. IBM CSECT eye-catchers are generally followed by an assembly date and a product identifier or PTF level, such as JBB3313 or UY01234; most eye-catchers are at the beginning of a module, but some are at the end.</li> <li>– In a module listed for the LPAMAP or VERBEXIT NUCMAP subcommand.</li> </ul> </li> </ul> <p>Subtract the starting address of the CSECT from the instruction address to obtain the offset into the CSECT.</p>	<p><i>IPCS Command Reference</i> for the subcommands:</p> <p style="padding-left: 40px;">LIST LPAMAP STATUS VERBEXIT LOGDATA VERBEXIT NUCMAP VERBEXIT SUMDUMP WHERE</p>
<p>9. <b>Collect and analyze any other problem data</b> recommended in the procedure for an abend in the <i>Basics of Problem Determination</i> book.</p>	

Diagnostic procedure	References
<p>10. <b>Report the problem to IBM</b>, if assistance is needed or if the problem is new. Provide the following problem data:</p> <ul style="list-style-type: none"><li>• Problem type: abend</li><li>• Search argument</li><li>• Dump formatted by IPCS, online or printed</li><li>• SDWAVRA keys, lengths, and contents</li><li>• Offset of the failing instruction into the module</li><li>• Accompanying messages: identifiers and texts</li><li>• SYS1.LOGREC report, if used</li><li>• Module name and level</li><li>• Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service</li><li>• Other problem data developed while using the <i>Basics of Problem Determination</i> book</li></ul>	<p>Chapter 4, “Reporting a Problem to IBM”</p>

---

## Diagnosing a Return Code Due to a Problem in Cell Pool Services

Use this procedure if the return code from the cell pool services part of callable service requests is one of the following. The codes indicate that cell pool services has found a problem in a data structure or in an associated service. The return codes are returned to the program issuing the call and appear in the caller's parameter list and register 15:

**Return  
Code  
hex(dec)**

**64 (100)  
68 (104)  
6C (108)  
70 (112)  
74 (116)**

The return codes may be caused by one of the following conditions:

- The application is reusing storage that was originally given to a cell pool.
- The memory management service that the application is using encountered a problem.
- There could be a problem with the code within the cell pool services part of the component.

Diagnostic procedure	References
<p><b>1. Obtain an SVC dump from the requester of callable services</b></p>	
<p><b>2. Collect information about cell pool from the dump using the IPCS CBSTAT and CBFORMAT subcommands.</b></p>	<p>Chapter 2, “Collecting and Analyzing Problem Data for Callable Service Requests” on page 2-1</p>
<p><b>3. Develop a search argument</b> consisting of:</p> <ul style="list-style-type: none"> <li>• Program identifier: PIDS/5752SCCSR</li> <li>• Return code: PRCS/hhhhhhhh</li> </ul> <p>Use the search argument to search problem reporting data bases. If the search finds that the problem has been reported before, request the problem fix. If not, continue with the next step.</p>	<p>Chapter 3, “Developing a Search Argument for Callable Service Requests”</p>
<p><b>4. Collect dump data.</b></p> <p>Format the dump and print it or store it in a data set. Use the following IPCS subcommands, in this order. If using IPCS interactively, respond yes to the IPCS message that asks if summary data can be used by dump access.</p> <pre> STATUS FAILDATA SYSTEM CPU REGISTERS DATA CONTENTION WORKSHEET SUMMARY TCBERROR SUMMARY FORMAT VERBEXIT LOGDATA VERBEXIT SUMDUMP VERBEXIT TRACE VERBEXIT MTRACE CBSTAT addr STRUCTURE(CSRCPOOL) CBFORMAT addr STRUCTURE(CSRCPOOL) VERBEXIT SYMPTOM           </pre>	<p><i>Using Dumps and Traces</i> for requesting and printing dumps</p> <p><i>IPCS Command Reference</i> for the subcommands</p> <p>Chapter 2, “Collecting and Analyzing Problem Data for Callable Service Requests”</p>
<p><b>5. Collect and analyze any other problem data</b> recommended in the procedure for a return code in the <i>Basics of Problem Determination</i> book.</p>	
<p><b>6. Report the problem to IBM</b>, if assistance is needed or if the problem is new. Provide the following problem data:</p> <ul style="list-style-type: none"> <li>• Problem type: return code</li> <li>• The search argument</li> <li>• Module name and level</li> <li>• Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service</li> <li>• Other problem data developed while using the <i>Basics of Problem Determination</i> book</li> </ul>	

## Diagnosing a Return Code Due to a Problem in Data Windowing Services

Use this procedure if the return and reason codes from the data windowing services part of callable service requests are the following. These code combinations indicate a problem in data windowing services. The return codes are returned to the program issuing the call and appear in the caller's parameter list and register 15. The reason codes are returned to the program issuing the call and appear in the caller's parameter list and register 0.

Return Code hex(dec)	Reason Codes (hex)
4 (4)	00000125
8 (8)	00000118, 00000119, 00000152

Diagnostic procedure	References
<p>1. <b>Develop a search argument</b> consisting of:</p> <ul style="list-style-type: none"> <li>• Program identifier: PIDS/5752SCCSR</li> <li>• Return code: PRCS/hhhhhhhh</li> <li>• Reason code: PRCS/hhhhhhhh</li> </ul> <p>Use the search argument to search problem reporting data bases. If the search finds that the problem has been reported before, request the problem fix. If not, continue with the next step.</p>	Chapter 3, "Developing a Search Argument for Callable Service Requests"
<p>2. <b>Collect and analyze any other problem data</b> recommended in the procedure for a return code in the <i>Basics of Problem Determination</i> book.</p>	
<p>3. <b>Report the problem to IBM</b>, if assistance is needed or if the problem is new. Provide the following problem data:</p> <ul style="list-style-type: none"> <li>• Problem type: return code</li> <li>• The search argument</li> <li>• Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service</li> <li>• Other problem data developed while using the <i>Basics of Problem Determination</i> book</li> </ul>	

---

## Chapter 2. Collecting and Analyzing Problem Data for Callable Service Requests

This chapter contains information to help you collect and analyze data from a dump for a problem in the cell pool services part of callable service requests.

---

### Using the IPCS CBSTAT and CBFORMAT Subcommands to Obtain Cell Pool Services Data

If a problem occurs, the program requesting the cell pool service requests a dump that includes cell pool data. This dump contains unformatted cell pool data. Use IPCS to format this data. For information about IPCS commands, see *IPCS Command Reference*, and for information about using IPCS, see *IPCS User's Guide*.

Formatted cell pool data can be obtained in one of the following ways:

- Do the following:
  1. Start an IPCS session.
  2. Select the COMMAND option from the IPCS PRIMARY OPTION MENU.
  3. Enter one of the following subcommands on the IPCS Subcommand Entry panel:

```
CBSTAT addr STRUCTURE(CSRCP00L)
CBFORMAT addr STRUCTURE(CSRCP00L)
```

where *addr* is the address of a cell pool. The application supplied this area; obtain the address from the application.
- An IPCS application can exist and be used in conjunction with the application that originally requested the cell pool services. For example, an application can call cell pool services and, because of a problem, request an SVC dump. Later, after the dump is loaded onto a system for analysis, the IPCS application can collect and format information using CBSTAT and CBFORMAT.

## Output from CBSTAT Subcommand

If the address used in the CBSTAT command points to a valid CSRCPOOL and there are no errors, a sample of what you might see is:

```
-----  
IPCS OUTPUT STREAM ----- LINE 0 COLS 1 78  
COMMAND ==> SCROLL ==> CSR  
***** TOP OF DATA *****  
STATUS FOR STRUCTURE(CSRCPOOL) AT 00006000 ASID(X'0021')  
  
Cell Pool - User name: POOL #4  
  
CSR10002I No errors were detected in CSRCPOOL data.  
***** END OF DATA *****
```

If the address used in the CBSTAT command does not point to a valid CSRCPOOL, then a sample of what you might see is:

```
-----  
IPCS OUTPUT STREAM ----- LINE 0 COLS 1 78  
COMMAND ==> SCROLL ==> CSR  
***** TOP OF DATA *****  
STATUS FOR STRUCTURE(CSRCPOOL) AT 00006010 ASID(X'0021')  
  
Cell Pool  
  
CSR10001I ASID(X'0021') 00006010 does not point to a cell pool anchor.  
  
Anchor: 00006010  
+0000 00000004 00006040 00000004 00000000 | .....- ..... |  
+0010 00000000 00000000 00000000 00000000 | ..... |  
+0020 00000000 00000000 00000000 00000000 | ..... |  
+0030 C3E2D9C3 D7C5C201 | CSRCPEB. |  
  
***** END OF DATA *****
```

In the output, the content of what was thought to be a valid CSRCPOOL is displayed in hexadecimal and EBCDIC.

### Output from CBFORMAT Subcommand

If the address used in the CBFORMAT command points to a valid CSRCPOOL and there are no errors, then the cell pool information is displayed in a summary form:

```
-----  
IPCS OUTPUT STREAM ----- LINE 0 COLS 1 78  
COMMAND ==> SCROLL ==> CSR  
***** TOP OF DATA *****  
  
Cell Pool - User name: POOL #4  
  
Expected number of extents      4  
Number of extents found        4  
  Active                        0  
  Inactive                      1  
  Not connected                 3  
  Error                         0  
  
Cell Size                       4  
  
Cells          Number of cells  Available cells  
  Active extents           0           0  
  Inactive extents        256         256  
  Total                   256         256  
  
***** END OF DATA *****
```

If the address used in the CBFORMAT command does not point to a valid CSRCPOOL, then a sample of what you might see is:

```
-----  
IPCS OUTPUT STREAM ----- LINE 0 COLS 1 78  
COMMAND ==> SCROLL ==> CSR  
***** TOP OF DATA *****  
  
Cell Pool  
  
CSR10001I ASID(X'0021') .00006010 does not point to a cell pool anchor.  
  
***** END OF DATA *****
```



## Chapter 3. Developing a Search Argument for Callable Service Requests

Search procedures and arguments are described in the *Basics of Problem Determination* book.

The following table lists alphabetically the symptoms for search arguments for this component. For the symptoms for a particular problem, see the procedure for that problem in Chapter 1, “Diagnosing Problems in Callable Service Requests”. The symptoms in each procedure are listed in the order that provides the most efficient search.

Description	Free Format Symptom	Structured Format Symptom
Module name	ccccccc For example: CSRWVIEW	RIDS/ccccccc For example: RIDS/CSRWVIEW
Program identifier	5752cccc For example: 5752SCCSR	PIDS/5752cccc For example: PIDS/5752SCCSR
Return code	rchhhhhhh For example: rc00000008	PRCS/hhhhhhh For example: PRCS/00000008
Reason code	hhhhhhh For example: 000011A	PRCS/hhhhhhh For example: PRCS/000011A
System abend code	abendhhh For example: abend0C4	AB/S0hhh For example: AB/S00C4
Other search arguments developed while using the <i>Basics of Problem Determination</i> book		



## Chapter 4. Reporting a Problem to IBM

The checklist in the following table identifies the data that IBM needs for a problem in callable service requests. Collect the data appropriate for the problem type before calling to report a problem.

Problem	Example
Problem type	Abend
Search argument from Chapter 3, “Developing a Search Argument for Callable Service Requests”	
SVC or stand-alone dump formatted by IPCS, online or printed, including, if appropriate: <ul style="list-style-type: none"><li>• Output from CBSTAT and CBFORMAT subcommands</li></ul>	
Offset of the failing instruction into the module	0AF
SYS1.LOGREC report	
SDWAVRA keys, lengths, and contents	
Messages, identifiers, and texts	
Module name and level	CSRWVIEW at PTF level UY00934
Name and level of the operating system(s) with a list of program temporary fixes (PTF) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service	JBB3313 (MVS/System Product 3.1.3) at PUT tape level T8804
Other problem data developed while using the <i>Basics of Problem Determination</i> book	



---

## Chapter 5. Callable Service Requests Functions

Callable service requests are callable interfaces that allow access to system services by programs written in high level languages (HLL). The two callable service requests functions are:

- Data windowing services
- Cell pool services

---

### Data Windowing Services

Data windowing services are part of the callable service requests component. Data windowing services allow HLL applications to use the data-in-virtual system services via a standard CALL. In addition to supporting data-in-virtual, data windowing services use hiperspaces.

A hiperspace is a range of up to 2 gigabytes of contiguous virtual storage. It does not require central storage backing; it is backed by expanded storage with the potential to spill into auxiliary storage. For further information on hiperspaces, see *SPL: Application Development – Extended Addressability*.

Data windowing services provides the caller with the following features:

- **Temporary data objects** - The caller can specify that hiperspace storage will be created for use as a data object.
- **New permanent data objects** - The caller can specify that a new linear data set will be created for use as a permanent data object.
- **Dynamic allocation of existing data objects** - Data windowing services dynamically allocate the data object to a system DDNAME.
- **Scroll area** - The caller can specify that hiperspace storage is to be used as a scroll area. A scroll area captures data without modifying a permanent object.

### Data Windowing Services Functions

The functions of data windowing services are:

- **Identify and Access** - Allow an application program to define a data object and indicate if the object can be updated or only read by data windowing services.
- **Unaccess and Unidentify** - Allow an application to cancel the identify and access request.
- **View Begin** - Establishes an association (mapping) between all or part of the data object and a virtual storage area where the caller can view the object data.
- **View End** - Removes the association established by **View Begin**.
- **Scrollout** - Saves changes made to temporary objects, or to permanent objects if a scroll area was requested. Changes made to a permanent object with a scroll area can be saved without modifying the permanent object at that time.
- **Save** - Writes changes to permanent data-in-virtual objects.
- **Refresh** - Resets the **View**. Changes made to the data object since the last time it was saved are discarded.

A HLL application requests these functions by a CALL statement.

For further information on window services, see *Application Development Guide*.

---

## Cell Pool Services

Bit mapped cell pool services are part of the callable service requests component. The callable services provided by cell pool can be called by an application program written in an HLL.

Cell pool services consist of service routines that manage user-supplied areas of virtual storage as a cell pool. A cell pool contains a number of fixed-size elements called **cells** and a control structure for cell pool management. Cell pool services run as an extension to the application and have the same execution environment and PSW key. Cell pool services also use any recovery environment set up by the caller.

### Cell Pool Services Functions

The functions of cell pool services are:

- **Build** - Formats a cell pool anchor data area for the cell pool. Each cell pool must have an anchor.
- **Expand** - Formats a cell pool extent data area, connects additional caller-acquired storage to the extent, and activates the extent.
- **Connect** - Connects caller-acquired cell storage to an extent.
- **Activate** - Activates the cell storage for an extent.
- **Deactivate** - Deactivates a cell pool extent.
- **Disconnect** - Disconnects an extent's cell storage and makes it unavailable to the cell pool manager.
- **Get** - Allocates a cell from the cell pool to the calling program.
- **Free** - Returns a previously allocated cell to a cell pool.
- **Query** - Retrieves information about a cell, cell pool, or cell pool extent.

An application requests these functions by a CALL statement.

For further information on cell pool, see *Application Development Guide*.

---

## Cell Pool Exits for IPCS

Cell pool provides IPCS support to perform analysis and generate output that can be used for problem determination. See Chapter 2, “Collecting and Analyzing Problem Data for Callable Service Requests” on page 2-1 for information about IPCS for cell pool.

---

## Callable Service Requests Modules

The cell pool services modules are named CSRPCccc.

The data windowing services modules are named CSRcccc.



---

## Index

### A

- abend unexpected by cell pool services problem
  - diagnosing 1-4
  - identifying 1-1
- abend unexpected by data windowing services problem
  - diagnosing 1-7
  - identifying 1-1
- abend 019 problem
  - identifying 1-1
- abend 08B problem
  - identifying 1-1
- abnormal termination
  - See abend xxx for the particular code
- analyzing problem data
  - in callable service requests 2-1

### C

- callable service requests
  - function of 5-1
  - problem diagnosis in 1-1
- CBFORMAT subcommand
  - output 2-1
- CBSTAT subcommand
  - output 2-1
- cell pool services user problem
  - identifying 1-2
- collecting problem data
  - in callable service requests 2-1
- CSRcccc 5-3
- CSRPCccc 5-3

### D

- data
  - for problems
    - See problem data
- data windowing services user problem
  - identifying 1-3
- diagnosing problem
  - in callable service requests 1-1

### F

- function
  - provided by callable service requests 5-1

### I

- identifying problem
  - See problem type

### M

- module
  - in problem data 4-1
  - name of, for callable service requests 5-3

### O

- operating system
  - name
    - in problem data 4-1

### P

- problem data
  - See *a/so* analyzing problem data, collecting problem data
  - for callable service requests 4-1
- problem type
  - identifying in callable service services 1-1

### R

- return code from cell pool services problem
  - diagnosing 1-10
  - identifying 1-1, 1-2
- return code from data windowing services problem
  - diagnosing 1-12
  - identifying 1-2

### S

- search argument
  - symptom for callable service requests 3-1
- stand-alone dump
  - in problem data 4-1
- SVC dump
  - in problem data 4-1
- SVC 99 problem
  - identifying 1-3
- symptom
  - See *a/so* search argument
  - abend code 3-1
  - callable service requests 3-1
  - for problem in callable service requests 1-1
  - module name 3-1
  - program identifier 3-1
  - return or reason code 3-1
- SYSMDUMP ABEND dump
  - in problem data 4-1
- system completion code
  - See abend xxx for the particular code

## Numerics

000 to FFF

See abend xxx for the particular code

MVS/ESA  
Component Diagnosis:  
Callable Service Requests

READER'S  
COMMENT  
FORM

LY28-1024-1

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

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

Possible topics for comment are:

Clarity    Accuracy    Completeness    Organization    Coding    Retrieval    Legibility

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

---

---

---

---

What is your occupation? \_\_\_\_\_

How do you use this publication? \_\_\_\_\_

Number of latest Newsletter associated with this publication: \_\_\_\_\_

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

**“Restricted Materials of IBM”**  
**All Rights Reserved**  
**Licensed Materials - Property of IBM**  
(Except for Customer-Originated Materials)  
©Copyright IBM Corp. 1988, 1989  
LY28-1024-1

S370-37

Reader's Comment Form

c  
u  
t  
a  
l  
o  
n  
g  
t  
h  
i  
s  
l  
i  
n  
e

Fold and Tape

Please Do Not Staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
Department D58, Building 921-2  
PO Box 950  
Poughkeepsie, New York 12602-9935



Fold and Tape

Please Do Not Staple

Fold and Tape

Printed in U.S.A.







Program Number  
5685-001  
5685-002

File Number  
S370-37

---

"Restricted Materials of IBM"  
Licensed Materials – Property of IBM  
LY28-1024-1 © Copyright IBM Corp. 1988, 1989

LY28-1024-01

