

SAMS:
Compress®



DB2 EDITION

Release 3.2
June, 2000

USER'S GUIDE



© 2000 Computer Associates International, Inc.
All Rights Reserved

Computer Associates International, Inc.
Storage Management Division
625 East Carnegie Drive, Suite 200
San Bernardino, California 92408-3510

01CM200P002-32

Proprietary/Trade Secret Notice

SAMS:Compress is undergoing constant revision, refinement, and expansion to include additional features. The current documentation is believed to be accurate and reflects the current version of the software. Important information not available when this book was published is contained in the Cover Letter that accompanied the distribution tape.

Warranty Disclaimer, Right to Revise

This documentation and the software it describes, constitute the proprietary, confidential, and valuable trade secret information of Computer Associates International, Inc., and is licensed either "AS IS" or with a limited warranty, as set forth in the applicable license agreement. NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE.

Computer Associates International, Inc., reserves the right to revise this publication from time to time and to make changes in the content hereof without the obligation to notify any person or entity of such revisions or changes. No part of this User's Guide can be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, for any purpose, without the express written permission of Sterling Software, Inc.

Use, duplication or disclosure by the Government is subject to restrictions as set forth in Sections 52.227-14 and 52.227-19 of the FAR Commercial Computer Software Restricted Rights Clause and/or the Rights in Technical Data and Computer Software Clause at DFAR 252.227.7013 as applicable.

© 2000 Computer Associates International, Inc., All Rights Reserved.

Printed in the USA

Customer Support

In the United States, contact:
Computer Associates International, Inc.
Storage Management Division
10811 International Drive
Rancho Cordova, California 95670
(800) 889-0226

Outside the United States, contact your local Computer Associates customer support representative.

Contents

	Contents	i
	Preface	1
CHAPTER 1	General Information	3
	Features	5
	Software Requirement	6
	Technical Support Procedures	6
CHAPTER 2	How SAMS:Compress Works	7
	Compression Routines	7
	Selection of Routines	9
	Compression Analysis Repository	10
	Compression Information	10
CHAPTER 3	ISPF Compression Analysis.....	11
	ISPF Panels	11
	Initial Panel	13
	Database/Creator/Table Specification Panel	14
	Database Selection Panel	15
	Table Selection List and Statistics Panel	16
	Compression Analysis Parameters Panel	19
	Batch Compression Analysis	21
	National Language Selection Panel	22
	Compression Analysis Detail Panels	22
	Selecting the Best Compression Routine	28

	Create Unique EDITPROC Panel	31
	Recreate Unique EDITPROC Panel	32
	Reported Results Panel	33
	Confirm Immediate EDITPROC Implementation Panel	35
	Implementation Considerations Panel	37
CHAPTER 4	Batch Compression Analysis.....	39
	Using the ANALYZE Command	40
	About the Batch Report	41
CHAPTER 5	Compression Analysis Repository	45
	Compression Analysis Repository Design	46
	Using the Repository	47
	Number of Repositories to Create	47
	Repository Data Access	48
	Deleting Rows and Dropping Tables	49
	Creating a Repository	50
	Repository Setup Panel	50
	Repository Specification Panel	51
CHAPTER 6	Implementing Compression	53
	Implementing a Compression Routine	53
	Immediate EDITPROC Implementation	55
	Using the Immediate EDITPROC Implementation Utility	55
	Preserving Your Data	57
	Restrictions on the Immediate EDITPROC Implementation Utility	58
CHAPTER 7	Using Additional Programs on Compressed Tables.....	59
	Add Column Utility	59
	Using the Add Column Utility	59
	Utility Commands	60
	Preserving Your Data	62
	Restrictions on the Add Column Utility	62
	Print Utility	63
	Using the Print Utility	63
	Restrictions on the Print Utility	64
	Third Party Program Capability	65
APPENDIX A	Installing	67
	Installation Tape Contents	67
	Installation Checklist	68
	Installation Steps	68

	Upgrade Considerations	73
	Third Party Program Installation Considerations	73
Appendix B	Messages	75
	Batch Programs Error Messages	75
	Online Error Messages	93
	Online Informational Messages	106
	Index	111

Preface

SAMS:Compress 3.2 has incorporated the following enhancements to the previous version:

- The product is no longer separated between Basic and Advanced functions. All of the functionality is now available from one ISPF panel structure. The change makes the operation and installation of SAMS:Compress for DB2 simpler and quicker.
- SAMS:Compress provides the ability to immediately implement compression on a table. When selecting this option, you do not need to drop and recreate the table. You simply need to unload and reload the table to entirely compress the table.
- SAMS:Compress now allows you to add columns to a table compressed with any of the new flexible routines.
- It compresses the Japanese Kanji Double Byte Character Set (DBCS).
- It lets you submit compression analysis to batch from within the ISPF panels.

Sterling Shrink for DB2 Version 3.1.0 incorporated the following enhancements to the previous version:

- Supports the Japanese Single Byte Character Set (SBCS).
- Supports DB2 Version 2.3 enhancements:
 - Accommodates the decimal column capacity in DB2 Version 2.3 that has been increased from 15 digits to 31 digits.
 - Accommodates the number of columns allowed in a table that has been increased in DB2 2.3 from 300 to 750.

Sterling Shrink for DB2 Version 3.0 incorporated the following enhancements to the previous version:

- General improvements to compression routines:
 - SnnINTEL (Intelligent Compression Routine)
 - Improved overall performance
 - Improved numeric compression and performance
 - Improved date/time/time stamp compression and performance
 - Improved NULL field performance
 - SnnXPRES (Express Compression Routine)
 - Improved overall performance
- Provides the following new compression routines that allow you to balance table compression with database performance:
 - SnnSHRVL—Additional compression for CHARACTER data types.
 - Optimal compression of all columns—An optimal version of our popular and powerful Intelligent Compression Routine using a fraction of the CPU requirements.
 - Optimal compression using COPE (Sterling Shrink's Compression Optimizer and Performance Enhancer facility)—Optimal compression routines generated specifically for your DB2 tables that substantially reduce the CPU requirements necessary to achieve high levels of compression.
- Improved compression analysis allows you to:
 - Specify the subsystem for analysis.
 - Specify database, creator, and table name patterns to use in defining the table analysis list.
 - View additional information detailing the amount of compression and relative performance of each of the Sterling Shrink compression routines.
- Sterling Shrink Repository saves the results generated during compression analysis, allowing you to query this information using SQL.
- National Language Compression incorporates foreign language compression tables within Sterling Shrink's Intelligent Compression Routine, ensuring the highest level of compression for your non-American based character tables.
- Sterling Shrink Print Utility provides you with a new utility that prints compressed pages in a similar manner to the IBM DSN1PRNT Utility.

General Information

IBM's DB2 provides a powerful relational database system that can be used for implementation of modest to massive database applications. However, DB2 does not provide efficient and effective ways to reduce DASD space by compressing table data.

As with other files and database systems, DB2 database tables contain columns that must be defined large enough to hold any valid value in each column.

For example, a column defined to hold a person's name must have the capacity for both "Tom Cox" and "Mr. Thomas Reginald Thrombus III." Because databases must accommodate rows and columns that vary widely in size, the use of space is extremely inefficient.

SAMS:Compress resolves the column-contents problem by compressing data to make the most efficient use of DASD space. When implemented for a DB2 table, SAMS:Compress transparently compresses all table row information before storing it on DASD. When DB2 reads the information back from DASD, SAMS:Compress automatically decompresses the information. Application programs and interactive SQL and DB2 utilities all function normally with SAMS:Compress.

SAMS:Compress is easy to implement and use. Consider the following points:

- **Implementation with standard MVS and DB2 system interfaces**—This implementation approach helps to ensure that a system's integrity and reliability is maintained on both current and future releases of DB2 and operating system software.
- **Minimal user input required with all ISPF functions**—For example, lists of items (databases and tables) are presented on the panels for user selection. The ISPF panels also include help panels that are accessible simply by pressing the ISPF HELP key.
- **No need for special interfaces to DB2 to perform compression**—SAMS:Compress uses the existing DB2 EDITPROC facility to intercept data before it is written to and after it is read from DB2 database tables. The SAMS:Compress EDITPROC routine automatically compresses data being written to a DB2 table and decompresses data once it is read from a DB2 table.
- **Step-by-step guidance through the compression analysis process**—SAMS:Compress reads the DB2 system catalog and presents you with selection lists of databases and tables that are candidates for compression, based on user selection criteria.
- **Implementation only for those DB2 tables specified**—You can implement compression selectively, weighing performance trade-offs against the benefits to be received from compressing data prior to implementation.
- **Immediate implementation of existing DB2 tables**—By using a flexible compression routine, you can implement compression without having to drop and recreate the table. The flexible compression routine also lets you add new columns to a compressed table.

Features

SAMS:Compress provides you with a comprehensive DB2 compression product. Its functions provide you with the capability of controlling precisely how much compression to impose on a table, how much overhead is required to obtain that compression, and how much flexibility the table retains during and after compression.

SAMS:Compress offers you the following features:

- **Forecasting of DB2 requirements**—This feature provides you with a means of determining anticipated compression benefits for a table that has not yet reached its expected size.
- **Transparent compression**—Compression occurs automatically as data is written to the DB2 table and decompression occurs after it is read, making the process invisible to users of the DB2 table.
- **Independence of the DB2 version**—This feature provides integrity for your data and ensures that your compressed data is always retrievable.
- **Extensive choice of compression routines**—You can select from seven different routines, selecting a compression routine that best balances your needs for database performance and DASD space saving. Compression routines available to you are described below.
 - **Intelligent compression of common data mixes**—This routine delivers acceptable compression with minimal overhead for most DB2 tables.
 - **Intelligent compression of irregular data mixes**—This routine provides more compression with a slight increase in overhead when compressing tables that contain irregular data mixes.
- **Optimal compression of all columns**—This routine is an optimized version of the intelligent routine created for use with a specific table. It provides the same amount of compression as the intelligent routine but with reduced overhead.
- **Optimal compression created by COPE (Compression Optimizer and Performance Enhancer)**—This expert system facility creates up to three optimal compression routines that provide significant reductions in overhead with differing amounts of compression. The three routines are:
 - A balanced routine that achieves moderate compression with acceptable overhead rates.
 - A routine that reduces the overhead level required by the balanced routine while still providing some compression.
 - A routine that increases the compression level achieved by the balanced routine while still minimizing the overhead requirements.
- **High performance Express compression**—This routine incorporates fast compression with minimal impact on performance.
- **Flexible compression routine**—SAMS:Compress provides the ability to immediately implement compression on a table. When this routine is implemented, you do not need to drop and recreate the table. You only need to perform a REORG to compress the entire table. The flexible

routines also allow you to add columns in the future.

- **National language compression**—These routines provide you with a choice of intelligent and flexible compression routines that have been optimized for compression of data of different foreign languages; resulting in increased compression with reduced overhead for national language data.
- **Compression analysis of any table under consideration for compression**—During this ISPF or batch analysis, SAMS:Compress reads and simulates compression of live data so you can view the amount of compression and relative performance of each compression routine as it would be implemented on the selected table. Analysis information is provided for both INSERT/UPDATE and SELECT processing operations.
- **Compression analysis repository**—SAMS:Compress stores the results of your compression analysis in a repository of SAMS:Compress-created DB2 tables. This feature allows you to perform analysis at different times and have the results available for reference through the SAMS:Compress ISPF dialog or standard DB2 SQL queries.

Software Requirement

SAMS:Compress has the following software requirements:

- Operates in MVS/XA and MVS/ESA environments.
- Functions with all versions of IBM's DB2 system.
- Uses IBM's ISPF TSO dialog facility for its ISPF compression analysis facility panels.

Technical Support Procedures

If you are having problems, perform as many of the following diagnostic steps as are relevant prior to contacting SAMS:Compress Technical Support.

- Read the documentation carefully.
- Reconstruct the events and write them down.
- Run some simple test cases.

If a problem arises that requires technical programming or analysis support, contact the SAMS:Compress Technical Support Center at the telephone number below.

- Telephone: (800) 889-0226

Do not ask for a specific support representative unless you are already working with someone on your particular problem.

You can transmit small numbers of pages to our facsimile machine by using the following telephone number.

- FAX: (916) 852-8601

How SAMS:Compress Works

To take advantage of the capabilities of DB2 requires the use of critical resources in your data center. These resources vary depending on the DB2 functions you are using and on the number of concurrent users.

SAMS:Compress offers significant benefits in helping data centers control the critical resource of DASD space when using DB2. In addition, SAMS:Compress also controls CPU utilization.

Using SAMS:Compress involves two basic steps. The first step is compression analysis, the process that assists you in determining which tables you should consider compressing. If your data center has specific concerns and requirements regarding the amount of DASD space and CPU time available, the compression analysis facility lets you determine the impact compression will have on each of these resources. You gain control over how much you reduce DASD space requirements and at what cost in additional overhead.

Note You do not have to analyze a table before implementing compression on it.

The second step is implementing compression for a selected table. After SAMS:Compress is implemented, it is transparent to all applications using the table. No programming changes are necessary to read or write data.

Compression Routines

SAMS:Compress provides you with a choice of three types of compression routines, enabling you to balance convenience of use against effective compression. Guidelines to help you select the appropriate compression routine for your tables begin on “Selecting the Best Compression Routine” on page 28. The following provides an explanation of the different compression routines available.

The three types of compression routines are:

- 1 Standard routines
- 2 Flexible routines
- 3 Optimal routines

The specific routines available within these three types are discussed below.

Standard Routines

Standard routines are those that are provided with SAMS:Compress. They contain compression algorithms to compress and decompress entire rows of data in different manners. These routines contain compression tables that are not changed.

The following standard routines are available:

SnnFLEX1	Gives you the ability to immediately implement compression on a table and lets you add new columns to a table that has already been compressed. Corresponding National Language versions of the flexible routine are available.
SnnINTEL	Provides the best compression and overhead available with a standard routine by using dynamic column-level algorithm selection.
SnnSHRVL	Provides more compression of character columns than SnnINTEL, with a slight increase in overhead.
SnnXPRES	Provides the least amount of overhead for a standard routine, with a decrease in the amount of compression.
Snntccsn	National language compression routines provide increased compression and reduced overhead for specific foreign languages.

Note: The letter *n* is used in the routine names to represent the release number of the SAMS:Compress product in which the routine was introduced. The letters *tccsn* represent the national language abbreviation, where *t* is either I—INTEL or F—Flexible.

Optimal Routines

Optimal routines are created by SAMS:Compress for optimally compressing specific tables. These routines are created only when requested by you for a specific table. Thus, each routine is unique for the table for which it is created. The compression tables in these routines are the same as those in the standard routines, which means that you do not risk losing the key to accessing your data when using an optimal routine.

Optimal routines are easily created using the ISPF interface. Two types of optimal routines are created. The first type created is simply an optimized version of the standard intelligent compression routine (SnnINTEL). Both SnnINTEL and its optimized counterpart compress every column in a table. The optimal version provides the same level of compression as the standard routine; however, it is optimized to use less overhead. Although the reduction in overhead depends on the make-up of the table under analysis, savings can approach twenty percent.

The second type created is not based on any standard routines and may not compress every column; however, these optimal routines still use the same compression tables. The routine composition is determined by

SAMS:Compress, using COPE (Compression Optimizer and Performance Enhancer expert system), part of SAMS:Compress Compression Analysis. Optimal compression is explained below.

Most DB2 tables contain columns of different types of data. Each of these columns is compressed with a specific algorithm, based on the data contained in the column. Some columns contain data that is highly compressible within a short amount of time. Others contain data that requires CPU time to achieve only a small amount of compression. COPE determines which columns can be compressed most efficiently to achieve certain thresholds of compression.

For example, assume that a table contains two columns (CHARACTER(50) and INTEGER). After compression analysis, SAMS:Compress determines that each column is compressed approximately fifty percent and that each uses the same amount of processing time for compression. The COPE facility favors compressing the character column because an equal amount of CPU processing time can reduce the row size by 25 bytes as opposed to 2 bytes for the integer column.

The DB2 practice of storing data by pages and placing complete rows only on those pages requires SAMS:Compress to compress columns effectively in order to place the greatest number of rows on a DB2 page. The COPE facility, therefore, selects columns to compress by balancing a column's compression efficiency, as explained above, against the need to place the greatest number of rows onto a DB2 page.

SAMS:Compress uses these results to build optimal routines, and includes them in the compression analysis. The results are presented, allowing you to determine if creating a unique EDITPROC provides justifiable benefits for specific tables.

The SAMS:Compress COPE facility determines the composition of one, two, or three different optimal routines and generates them for use in compression analysis and unique routine creation. These routines are described below.

Note The COPE facility assumes that the table under analysis does not share its tablespace with another table.

- A balanced routine achieving moderate compression with acceptable overhead rates. The routine compresses only the columns that provide significant compression in an average amount of time.
- A routine that reduces the overhead required by the balanced routine while still providing some compression. This routine compresses only those columns that can be compressed quickly.
- A routine that increases the compression achieved by the balanced routine while still minimizing the overhead requirements. This routine compresses those columns that have the potential for the highest percentage of compression.

Selection of Routines

SAMS:Compress provides a comprehensive analysis facility that helps you to quickly determine the benefits of different compression routines. The SAMS:Compress ISPF interface performs compression analysis and reports to you the amount of compression and relative performance of each of the

SAMS:Compress compression routines. Using this information, you can rapidly determine which routine provides you with necessary reductions in DASD space with an acceptable amount of overhead.

Compression Analysis Repository

The SAMS:Compress Compression Analysis Repository contains the results of previous analyses. There are several benefits in having the repository. With the repository, you can perform analyses at different times, for different tables, and retain the results of those analyses for future consideration regarding which tables to compress and which routines to use. You can view the results using the SAMS:Compress ISPF interface, or develop your own queries using SQL or QMF. Finally, the repository archives analysis information that is used in creating optimal routines, allowing you to recreate a routine if necessary.

Compression Information

SAMS:Compress compresses all characters in each row of DB2 table data. Since DB2 does not use key information stored in each row to access table data, key data can also be compressed, thereby allowing maximum use of DASD space. Since DB2 does not store actual index information in table rows, SAMS:Compress compression does not need to avoid compression of indexes. SAMS:Compress evaluates each row to determine if it is a candidate for compression. Sometimes rows contain data that is not compressible. Other times the amount of compression realized from compressing a row does not justify the time required to perform the compression. For these reasons, occasionally a row of DB2 data is not stored in compressed format. Thus, a DB2 table selected for compression can consist of compressed and uncompressed rows of data. A SAMS:Compress uncompressed data identifier appears in each uncompressed row. When this identifier is found in a row, the row is uncompressed.

Each row of compressed data contains the following information:

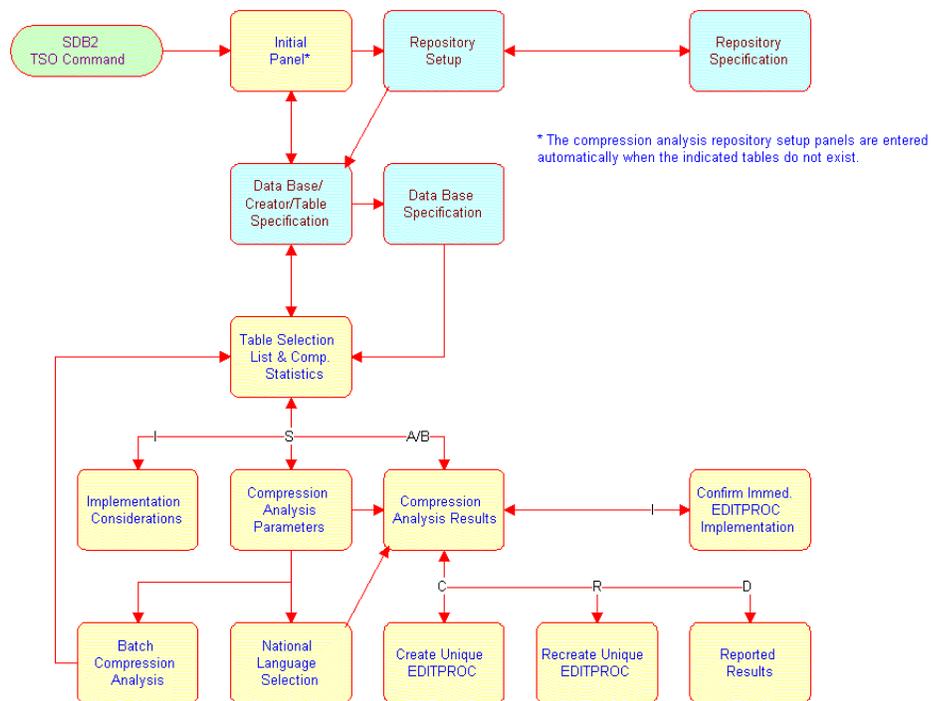
Compressed Data	The actual compressed data.
DB2 Control Information	SAMS:Compress leaves intact any DB2 control information that exists in each row.

ISPF Compression Analysis

ISPF Panels

The SAMS:Compress ISPF interface performs compression analysis and reports to you the amount of compression and relative performance of each of the SAMS:Compress standard and optimal compression routines. Compression analysis panel flow is shown in Figure 1.

FIGURE 1. ISPF Compression Analysis Panel Flow



Each of these panels is described in detail below.

Initial Panel	The first panel lists the DB2 subsystem and the repository ID that you use for compression analysis. You can change either the subsystem or repository ID.
Repository Setup Panel	This panel specifies the name of the compression analysis repository to be created.
Repository Specification Panel	This panel allows you to identify the storage group, database, or tablespace where the new repository tables are to be located.
Database/Creator/ Table Specification Panel	The specification panel lets you define which databases are to be included on the database selection list, or which tables are to be included on the table selection list.
Database Selection Panel	The names of all DB2 databases defined to the system that match the pattern specified on the specification panel are displayed on this panel. Use this panel to select a database to use in creating the table selection list.
Table Selection List and Compression Statistics Panel	This panel displays the tables that match the selection criteria from the specification or database selection panels along with their initial statistics. Use this panel to select a table for compression analysis.
Compression Analysis Parameters Panel	Before compression analysis begins, specify the parameters to control the amount of data to process and the amount of time to allow for the analysis.
Batch Compression Analysis Panel	This panel lets you submit a job to batch which will perform the compression analysis of a table.
National Language Selection Panel	This panel lets you select national language compression routines for use in analysis and displays the EDITPROC names.
Compression Analysis Results Panel	This panel displays the results of current and previous compression analyses. Use this panel to determine which compression routine to implement for a given table.
Create Unique EDITPROC Panel	This panel allows you to name a unique EDITPROC prior to implementing an optimal compression routine on a table.
Recreate Unique EDITPROC Panel	This panel lets you select any previously created unique routine to recreate.
Reported Results Panel	This panel shows additional detailed results of compression analysis for the indicated routine.

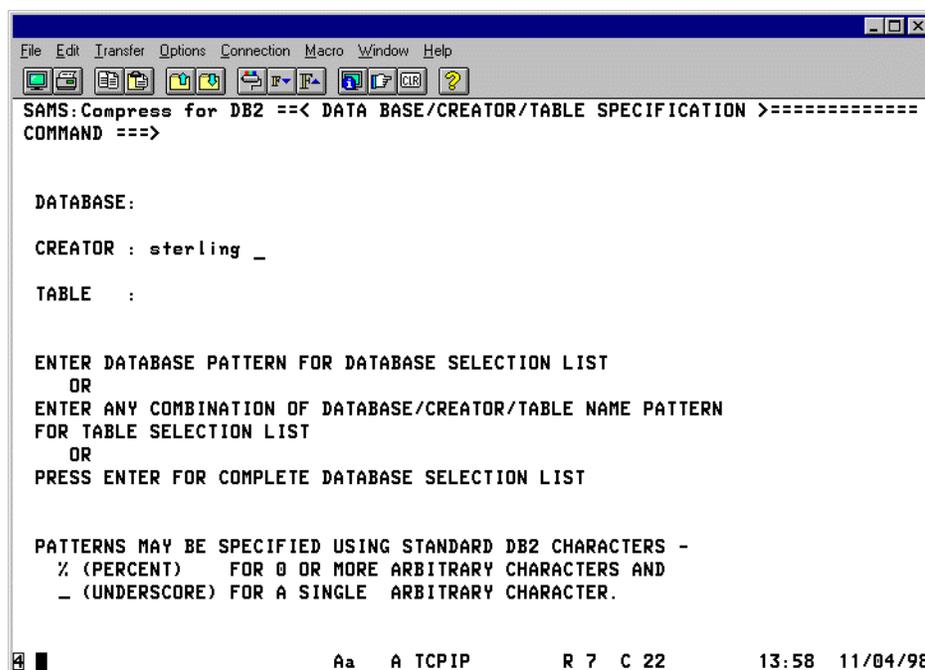
The identification of the repository is done by using a combination of creator ID and prefix. To specify the repository name, move the cursor to the Creator field and enter from 1 to 8 characters for a creator ID. Move the cursor to the Prefix field and enter from 1 to 8 characters for the prefix. Press <ENTER> to continue. SAMS:Compress allows you to have as many different repositories as you want. A single repository can be used for consolidating information in a single place. The implications of having multiple repositories are discussed in “Using the Repository” on page 47.

If the repository tables referred to by the combination of creator and prefix have not been created yet, SAMS:Compress automatically presents the Repository Setup Panel (Figure 19, “Repository Setup Panel,” on page 50), allowing you to control the creation of the repository.

Database/ Creator/Table Specification Panel

The Database/Creator/Table Specification Panel allows you to specify which tables to list as candidates for compression analysis. You may specify any combination of database, creator, and table name patterns to cause the Table Selection List and Compression Statistics Panel to display the tables that match the specified patterns. Or you can specify only a database name pattern for SAMS:Compress to use in displaying the Database Selection List Panel. If the name of the database containing the table you want to analyze is unknown, press <ENTER> to get a list of all databases in the current DB2 subsystem.

FIGURE 3. Database/Creator/Table Specification Panel



The patterns are specified using standard pattern recognition characters. These are:

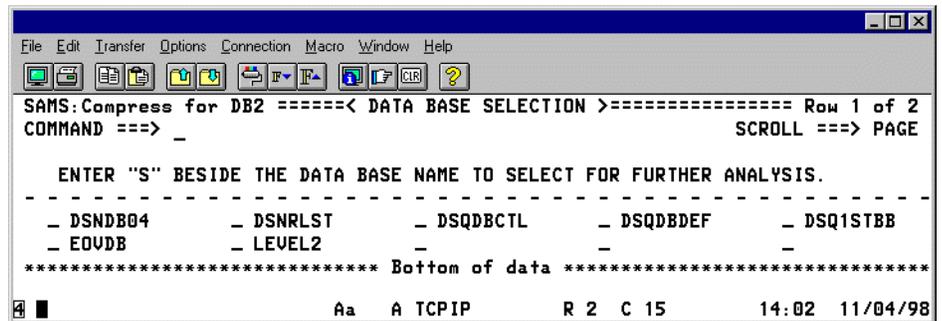
- % Any number of characters, including none
- _ Any one character for each "_"

In the panel shown above, any table that has a creator ID starting with "STERLING" and a table name starting with "NAME" is displayed on the Table Selection List and Compression Statistics Panel.

Database Selection Panel

The Database Selection Panel lists all user databases defined in the DB2 catalog that match the selection pattern specified in the Database field on the Database/Creator/Table Specification Panel. If more than a screen full of database names are available, scroll through the screens using normal ISPF scroll functions.

FIGURE 4. Database Selection List Panel

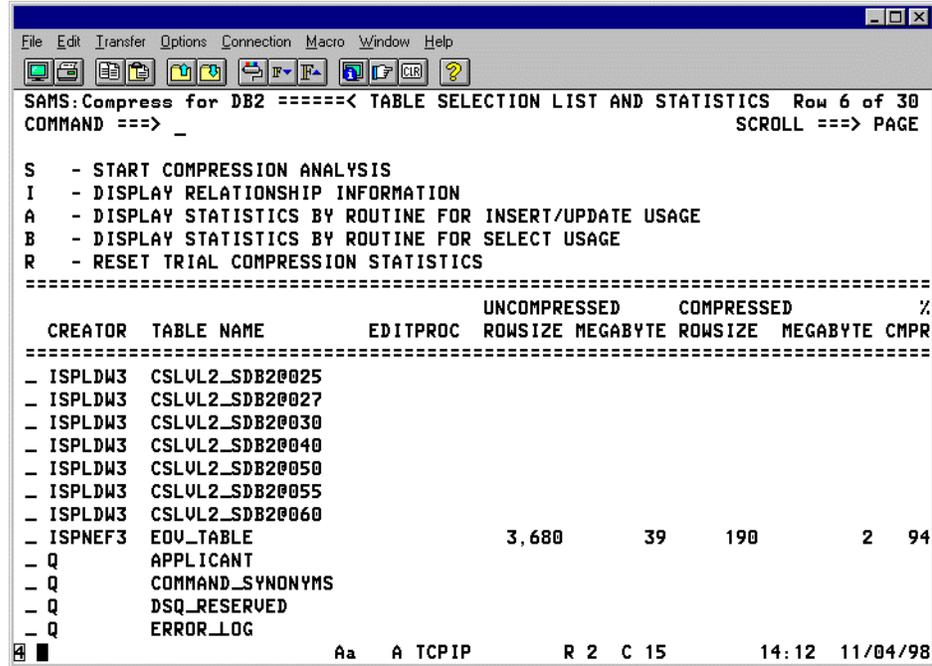


Select a database to list all of its tables by entering an S next to the database name and pressing <ENTER>. SAMS:Compress shows the panel shown on the next page, a list of tables in the database.

Table Selection List and Statistics Panel

This panel displays all tables that match the selection criteria specified on the Database/Creator/Table Specification Panel or all the tables contained in the database selected from the Database Selection Panel.

FIGURE 5. Table Selection List and Statistics Panel



The panel shows you the potential benefits of SAMS:Compress compression, lets you select compression analysis, view additional statistics for a given table, or perform a reset (clear) of the statistics in the SAMS:Compress repository.

The first field in the body of the screen is the blank selection field. To initiate a function, move the cursor to the selection field for the table name and enter one of the following:

Value	Definition
S	Starts the compression analysis simulation of the table. You are first prompted for control information concerning the analysis.
I	Displays a panel of the implementation considerations for the corresponding table. The resulting panel shows the number of DB2 objects (authorizations, plans, views, and so on) associated with the table. The information is presented because when implementing a non-flexible routine, you must recreate these DB2 objects.
A	Displays compression analysis results from a perspective of adding or updating rows to the table. Use this selection if the primary activity for the table is Insert/Update.

Value	Definition (Continued)
B	Displays compression analysis results from a perspective of retrieving rows from the table. Use this selection if the primary activity for the table is Select.
R	Resets any compression statistics that exist in the repository for the table. The use of reset allows you to delete information that is not needed in the repository.

The remainder of the screen shows a statistical account of the benefits of compression for the tables. This information initially appears if it exists in the repository. Otherwise, it is added as compression analysis is performed for each of the tables on the display. The following fields appear.

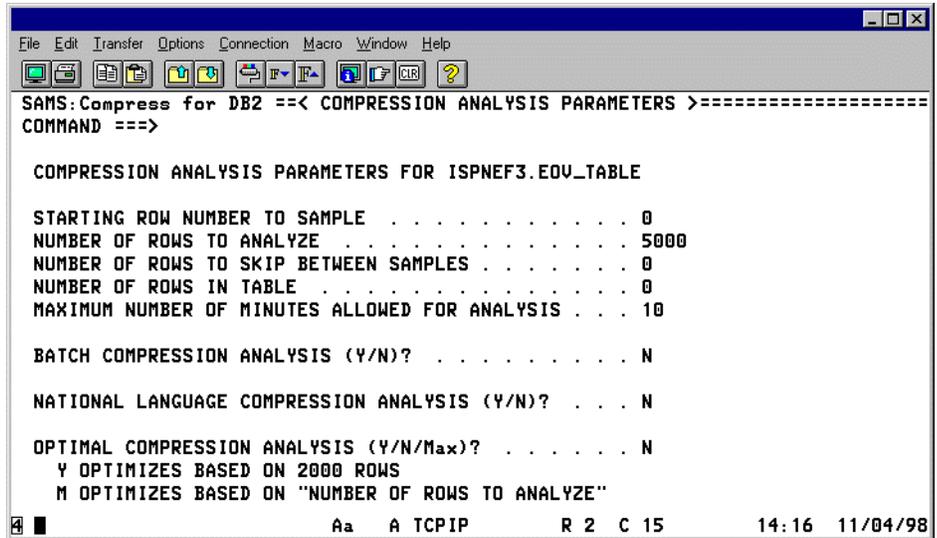
Field	Description
Creator	The creator ID of the table.
Table Name	The name of the table.
EDITPROC	<p>SAMS:Compress compression is implemented with the EDITPROC parameter on the DB2 CREATE TABLE statement. The EDITPROC statement refers to a routine that processes data before it is written to and after it is read from a DB2 table. This field indicates whether an EDITPROC routine already exists. This routine may be a user routine name or an existing SAMS:Compress compression routine.</p> <p>After SAMS:Compress has been implemented, this field shows which compression routine is being used on each table. You can use this screen to determine how SAMS:Compress is implemented at your installation.</p>

Field	Description (Continued)
Uncompressed	This field contains two types of information about the uncompressed table:
Rowsize	The average compressible row length for a table is displayed after compression analysis has been performed for that table. The rowsize excludes the 6 bytes at the beginning of each row that are required by DB2. Initially this value is blank, indicating that no compression analysis has been performed for the table.
Megabyte	<p>The number of megabytes of DASD required to accommodate the table is calculated and displayed when the panel is initially shown. This figure allows you to quickly determine which tables can provide the greatest savings if compressed. For these tables, you can obtain the compression information by performing compression analysis.</p> <p>The information for this field is initially provided by using statistics from the DB2 catalog. If the RUNSTATS Utility has not been run for this table the field is blank.</p> <p>After compression analysis, this figure is based on information entered on the Compression Analysis Parameters Panel and calculated during compression analysis.</p>
Compressed	<p>This field contains three types of information. The information is displayed only after compression analysis has been performed for an individual table. The results of each analysis are displayed on this panel to allow comparison of compression results for the various tables displayed.</p> <p>The compression analysis information displayed is based on the routine that produced the highest level of compression for each table. Enter B next to a table to determine the name of this routine.</p>
Rowsize	This figure represents the average length of the compressed rows of the table.
Megabyte	This figure represents the number of megabytes of DASD that would be required to contain the compressed table.
% Cmpr	This figure represents the percentage of DB2 compression for a table that can be expected when the table is compressed.

Compression Analysis Parameters Panel

This panel lets you specify the parameters that control the compression analysis process. You are provided with defaults for these values and can accept or change the default information.

FIGURE 6. Compression Analysis Parameters Panel



The compression analysis parameters are the following:

- Starting Row Number to Sample** Enter the row number from which the compression analysis process is to begin. A value of 1,000 skips the first 999 rows in the table before beginning analysis.
- Number of Rows to Analyze** Enter the total number of rows SAMS:Compress is to analyze. This value does not include the number of rows to be skipped.
Note: If "0" is entered, all rows are analyzed.
- Number of Rows to Skip Between Samples** Specify the number of rows to skip. For example, to analyze every tenth row of a table, enter the digit 9 in this field.
Note: The first row is not used for compression analysis, but is used to build EDITPROC Parameters only.

**Number of Rows
in Table**

If entered, this value is used in calculating the megabytes of DASD required for both the uncompressed and the compressed table. If a value is not entered, the number of rows sampled are used in the calculation.

Sometimes you need to do some preparation before entering a value for the Number-of-Rows-in-Table parameter. If the figure created by RUNSTATS is out of date, you cannot obtain an accurate calculation of your DASD savings. You might choose to run RUNSTATS before doing trial analysis or actual compression.

However, you might find that forecasting (an estimate of the number of rows) is more suitable for you. If you have a large database, you may not want to take the time needed to perform RUNSTATS. Therefore, a fairly accurate estimate might be preferable.

Forecasting can also be helpful if you have a growing database or a shrinking database. Your best estimate of the anticipated number of rows might be your only way to anticipate how much space you will actually save when your table is put into production. Therefore, you may want to enter your carefully considered prediction in this field.

**Maximum
Number of
Minutes Allowed
for Analysis**

Compression analysis is normally very fast. For example, a table containing 10,000 rows could be analyzed in less than a minute. However, if the elapsed time allotted for compression analysis must be limited, enter the maximum number of minutes to be allowed for the process here. If compression analysis of a table is not finished within the time specified, the analysis stops and statistics are shown for the number of rows sampled so far.

**Batch
Compression
Analysis**

This option is used to submit the analysis to run in batch. If you select this option, the analysis results are not generated online and are not available until after the batch job is completed.

National Language Compression Analysis

If you want to view a list of the national language compression routines available for use, enter a Y in this field and the National Language Compression Panel is displayed before compression analysis.

Optimal Compression Analysis

If you want SAMS:Compress to generate and include optimal routines in the compression analysis process, enter a Y or M in this field. Otherwise, enter N.

The generation of optimal routines requires additional time but offers you a wider choice of routines to select for implementation of table compression.

To limit the amount of processing time, enter a Y in this field. This confines the COPE facility analysis to only the first 2000 rows in the sample. Otherwise, enter an M to include all the rows specified in the sample.

Batch Compression Analysis

This panel is used to supply the JOB card to your batch job. The four lines shown on the panel are placed at the beginning of the batch job. Enter the necessary information to execute the batch job.

FIGURE 7. Batch Compression Analysis

```

SAMS:Compress for DB2 =====< BATCH COMPRESSION ANALYSIS >=====
COMMAND ==> _

BATCH COMPRESSION ANALYSIS SUBMISSION FOR ISPNEF3.EDV_TABLE

ENTER SUBMIT TO SUBMIT BATCH COMPRESSION ANALYSIS JCL
ENTER EDIT TO EDIT BATCH COMPRESSION ANALYSIS JCL
ENTER BROWSE TO BROWSE BATCH COMPRESSION ANALYSIS JCL
ENTER END TO CANCEL BATCH COMPRESSION ANALYSIS

NOTE: RESULTS WILL NOT BE AVAILABLE UNTIL YOUR NEXT
ANALYSIS SESSION AFTER THE BATCH JOB COMPLETES.

JOB STATEMENT INFORMATION (REQUIRED FOR BATCH COMPRESSION ANALYSIS)
==>
==>
==>
==>

Aa A TCP/IP R 2 C 15 14:22 11/04/98

```

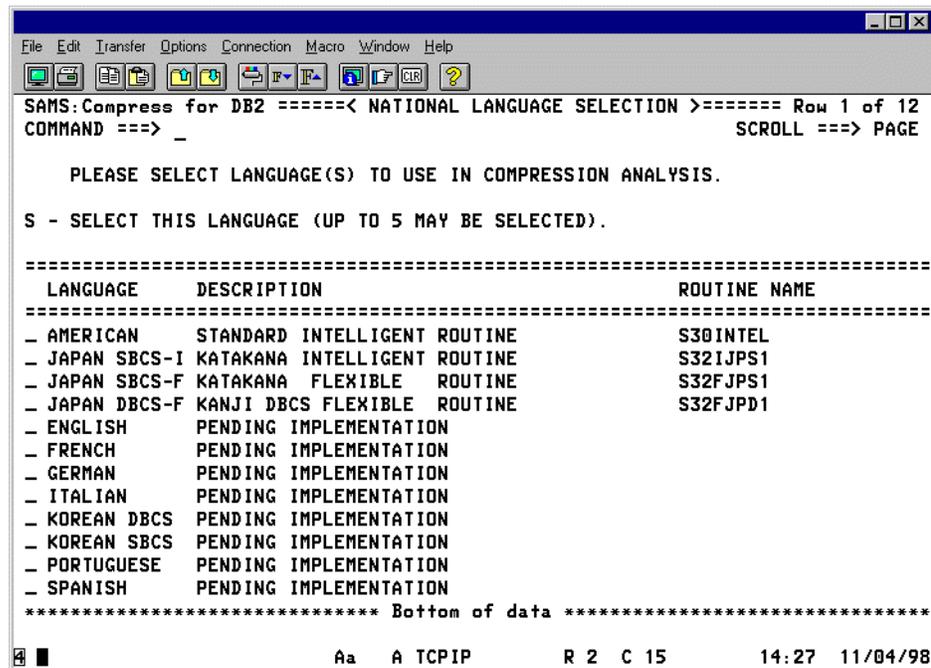
You can also use this panel to edit, browse, and submit the batch compression analysis JCL. The EDIT and BROWSE commands are used to access the entire JCL that will be submitted. When entered, you are placed into the standard SPF environment providing all of the edit/browse functions. This screen displays all of the compression analysis parameter statements. For an explanation of the parameters, see “Batch Compression Analysis” on page 39.

To execute the batch compression analysis, enter SUBMIT in the command line. The job is submitted and you are returned to the Table Selection List and Compression Statistics Panel.

National Language Selection Panel

SAMS:Compress includes several compression routines that have been optimized for compression of different national languages (foreign languages). This panel displays the compression routine names for each of the national languages currently incorporated into SAMS:Compress.

FIGURE 8. National Language Selection Panel



Those routines not fully incorporated are marked “Pending Implementation.” If you want to use a routine that is implemented, enter an S next to the routine and press <ENTER> to begin compression analysis using that routine. If you do not want to use any of the routines, press <END> and compression analysis starts.

The name of the compression routine that must be specified for implementing compression for the selected national language is displayed on this panel next to the language ID.

Compression Analysis Detail Panels

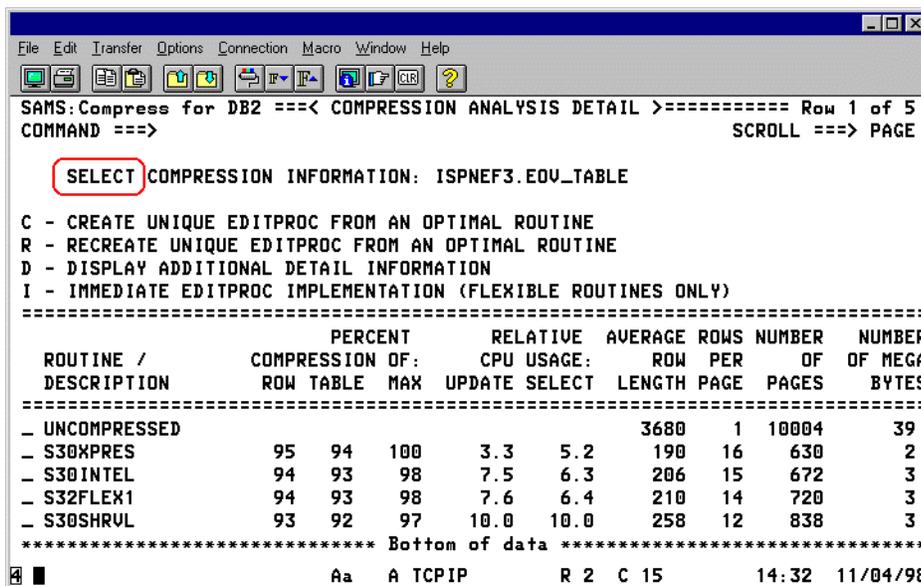
During compression analysis, information messages appear. After compression analysis is completed, SAMS:Compress displays the results of the analysis. Two separate panels contain the results of compression analysis for a selected table. The information on these panels allows you to select the best compression routine for the selected table, one that meets the

requirements you have for compression and overhead. The Compression Analysis Detail (Select) Panel shows compression information that assumes the table is primarily used for select operations. The Compression Analysis Detail (Insert/Update) Panel shows compression information that assumes the table is primarily used for insert/update operations. This panel is reached from the Table Selection List and Compression Statistics Panel.

The difference in these two perspectives results from the composition of the optimal routines created by COPE. The standard, flexible, and optimized intelligent routines are the same on the two screens. Since compressing columns can take a different amount of time than decompressing, the COPE facility can pick different columns to use in creating the various routines. This feature of the COPE facility allows you to select an optimal routine that is tailored for how you most often use the table.

The "Select" panel is automatically displayed when the analysis process has finished or after you enter a B on the Table Selection and Compression Statistics Panel.

FIGURE 9. Compression Analysis Detail (Select) Panel

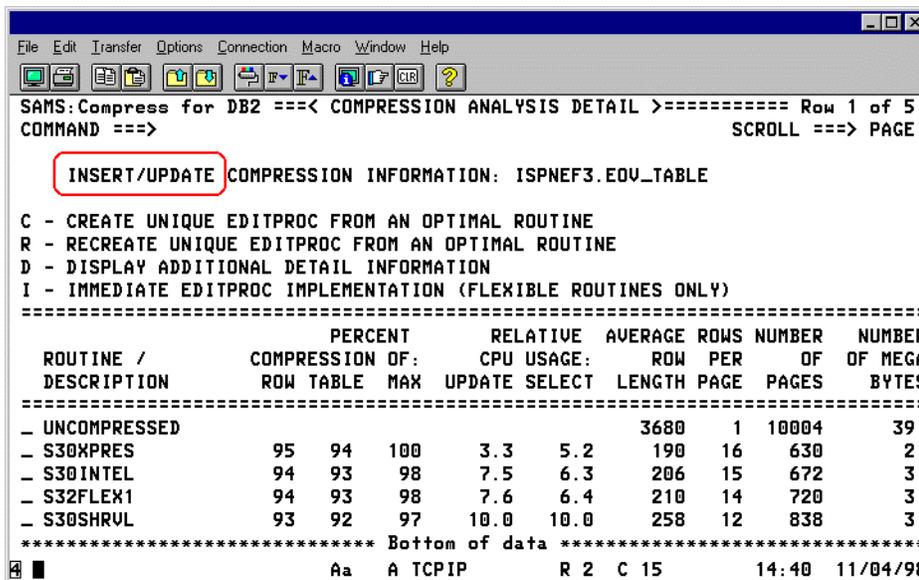


The table below shows a partial list of the DB2 operations characterized by the Select and Insert/Update panels. Since the fields on each panel are identical, they are discussed together.

Select Operations		Insert/Update Operations
BUILD	FETCH	INSERT
RUNSTATS	SELECT	DELETE
SORT	UNLOAD	UPDATE
		LOAD

The “Insert/Update” panel is displayed after you enter an A on the Table Selection and Compression Statistics Panel.

FIGURE 10. Compression Analysis Detail (Insert/Update) Panel



The panels in Figure 9 and Figure 10 show the potential benefits of SAMS:Compress compression for a single table. You can select to create a unique EDITPROC, recreate a unique EDITPROC, view additional compression statistics for a given routine, or immediately implement an EDITPROC.

The first field in the body of the screen is the blank selection field. To initiate a function, move the cursor to the selection field for the compression routine desired and enter one of the following:

Value	Definition
C	Creates a unique EDITPROC from one of the optimal routines. This function is used only for the optimal routines.
R	Recreates a unique EDITPROC. This function is used only for the optimal routines.
D	Displays a panel of additional compression statistics about the table generated during the analysis process.
I	Implements a flexible compression routine.

The remaining fields in the panels show a statistical account of the benefits of SAMS:Compress compression for the table.

**Insert/Update or
Select
Compression
Information**

The first line in this panel indicates the perspective view that the information displayed on the panel contains. The view is either INSERT/UPDATE or SELECT. Values differ only for the optimal routines in each of the two view perspectives. View the Compression Analysis Detail Panel that corresponds to how the table is primarily used.

**Routine
Description**

This field identifies each compression routine included in the compression analysis. Each line contains information that corresponds to the routine listed in this field.

The first entry contains information about the uncompressed table (UNCOMPRESSED). The other routines are displayed depending on the level of compression they achieved.

The SAMS:Compress standard routines (SnnINTEL, SnnSHRVL, SnnXPRES, and SnnFLEX1) are always displayed on the panels.

If you selected a national language compression routine, its results are displayed.

If you requested that optimal routines be calculated on the Compression Analysis Parameters Panel, the following types of optimal routines appear.

Optimized COPE Routines	Balanced, Less CPU, and More Compression. (All three routines are not always be created.)
-------------------------	---

Optimized Intelligent Routine	All Columns.
-------------------------------	--------------

**Percent
Compression of**

Percent compression shows two different types of comparisons. The first two values indicate a percent of compression in relation to the uncompressed table. The last value indicates the percent of compression achieved relative to the routine that produced the highest rate of compression.

Row This value is the amount of row compression obtained. Row compression is a figure that represents the average compression of all the analyzed rows in the table.

Table This value is the amount of compression obtained for the entire table. It indicates the amount of actual DASD space you can anticipate saving if compression is implemented on the table using this routine.

Max This value is based on the compression results shown in the previous column. The routine that achieved the highest level of compression is automatically set to one hundred percent and is displayed at the top of the routine list. The remaining values are set to the routines' percent difference from this best routine, and are displayed in descending order based on the MAX value.

Use this field to see how much compression a routine provides in comparison to the routine that achieved the highest level of compression, and to determine the choice of a routine for reducing CPU requirements while still obtaining an acceptable amount of compression.

Relative CPU Usage:

Relative CPU usage indicates how well (in terms of CPU requirements) the routine performed against the other routines listed on this panel. The routine that used the greatest amount of CPU processing time to compress the sample is assigned a performance value of 10.0. Each of the remaining routine's values should be evaluated in relation to this highest value. For example, if a routine had a value of 5.0, it used half as much CPU processing time as the highest valued routine to achieve its level of compression.

Update This value represents the routine's relative performance when performing Insert/Update operations.

Select This value represents the routine's relative performance when performing Select/Unload operations.

Note: The accuracy of these values can be adversely affected by the operational environment. The amount of time used to analyze a table is affected by the use of cache memory, extended storage, concurrent processing jobs, and PR/SM (LPAR) and VM environments. Each of these factors can cause the recorded simulation time to vary by as much as twenty-five percent.

Here is an example that shows how operating in a PR/SM environment can skew the results:

SAMS:Compress records the start time at the beginning of a simulation. If the process is interrupted, SAMS:Compress does not include that time in the final analysis. However, in PR/SM, if another LPAR obtains control (an IBM-defined "processor stall"), the active process is stalled in midstream and no indication is passed back to the LPAR. Since SAMS:Compress is not aware of this pause, the stall time is included in the final analysis. This event may potentially result in incorrect figures, depending on how often it occurs during the analysis.

Algorithms are incorporated within the analysis to reduce the potential for inaccurate results, but in some cases the results may need to be verified with multiple analysis runs. To improve the accuracy of the results for a single analysis, you can analyze a large sample of the table (the larger the better) or perform the analysis when activity on the machine is low (consider all LPARs in a PR/SM environment).

Average Row Length

The average row length of the compressed row for the routine is displayed.

Rows per Page	The number of rows that fits into a page of DB2 data after the rows were compressed is displayed. The number of rows per page is significant in determining the amount of compression that DB2 realizes when the table is compressed.
Number of Pages	<p>The number of pages that are required to contain the table after compression is displayed. This figure is based on one of the following numbers.</p> <ol style="list-style-type: none"> 1 The number entered in the Number of Rows in the Table field on the Compression Analysis Parameters Panel. This number may not represent the actual compression achieved if this table occupies the same tablespace as another table. 2 The number of rows in the table as reported by the DB2 catalog. This number SYSIBM.SYSTABLES.CARD is stored by RUNSTATS. 3 The number of actual rows sampled.
Number of Megabytes	The number of megabytes of DASD that are required for the compressed table is displayed. This figure is based on the value entered in the Number of Pages field and the DB2 page size for the table. This value may also be used to determine the correct allocation of DB2 for containing the compressed table. The number may not be accurate if this table occupies the same tablespace as another table.

Selecting the Best Compression Routine

SAMS:Compress assists you in selecting the best compression routine for implementing compression on a given table by supplying you with the necessary data. The information provided on the Compression Analysis Results Panels is sufficient to assist you in selecting the “best” routine. The following discussion is provided to help you through this process.

The first step in selecting the best routine is to determine the answers to the questions below:

- 1 How important is maximum compression for this table?
- 2 What is your acceptable amount of compression?
- 3 How important is the performance of DB2 when accessing the compressed table?
- 4 How is this table used the majority of the time (Insert/Update or Select operations)?
- 5 How important is it to use only a standard routine (not an optimal routine) for this table?
- 6 How much table flexibility in terms of ease of implementing compression and adding new columns in the future is needed?

Each of these questions is addressed in the following subsections.

Selecting a Routine for Maximum Compression

The selection of a routine for maximum compression is straightforward. Since the routine list is ordered by the amount of compression achieved, simply select the first routine. This routine produces the maximum compression for the table. (Percent Compression of: MAX equals one hundred percent.) It is a good idea to examine the next routine on the list to see if it achieves close to the maximum compression with a significant reduction in performance. For example, using the values shown in Figure 9, "Compression Analysis Detail (Select) Panel," on page 23, the second listed routine provided ninety-four percent of the compression with 2.4 units less of CPU requirements for Select operations.

Note This evaluation can also be performed using the Number of Megabytes column, which can give you more meaningful data to evaluate.

Selecting a Routine That Achieves an Acceptable Level of Compression

You can require that a selected table provide only a specific amount of compression, generally a low number. In this case, selecting a compression routine is again straightforward. Review the results panel under the Percent Compression of: Table and select the routine on the bottom of the list. If this value satisfies your needs, select this routine for implementation. This routine usually gives you the lowest CPU rates.

Selecting a Routine for Improved Performance, With Higher Levels of Compression

Selecting a routine that offers improved performance while still achieving an acceptable amount of compression is a process that requires you to carefully consider the information provided on the Compression Analysis Results Panels.

First, compare the information under the Percent of Compression: MAX with that in the Relative CPU Usage, under the Update or Select heading, depending on which is more critical for this table. The selection of the routine is based on how much weight you give to percent of compression and how much you give to the routine's relative performance. The best way to approach the selection is to start with the first routine and note that routine's maximum compression and relative performance figures. Next, note the second routine's corresponding maximum compression and relative performance. Compare the two sets of values and determine whether the decrease in relative performance values is high enough to outweigh the decrease in compression.

Selecting a Routine to Correspond to Its Primary Usage

If a table is used primarily for either Select or Insert/Update operations, you select the best routine using the routine list available from the proper view perspective panel.

The first step in this process is to select the correct view perspective, that of INSERT/UPDATE or SELECT from the Table Selection List and Compression Statistics Panel. To alternate between the Insert/Update and Select compression information perspective panels, press <END> to return to the Table Selection List and Compression Statistics Panel, then enter either an A or B next to the appropriate table to display the indicated compression information panel.

The second step is to identify the routine that provides the best mix of performance values within the selected view perspective.

Selecting an Optimal Routine

To implement an optimal routine used in the compression analysis, you must create a unique EDITPROC from that optimal routine. Creating a unique EDITPROC is a simple and quick process. SAMS:Compress provides an ISPF interface panel that allows you to name and enter a description for the routine. SAMS:Compress then creates the routine and places it in a designated load library for use as an EDITPROC.

During the process of creating a unique EDITPROC, SAMS:Compress archives the information that is necessary to recreate the EDITPROC (if that should ever be necessary). This feature, coupled with adequate backup procedures, ensures that you always have access to your data when using a unique routine.

Selecting a Flexible Routine

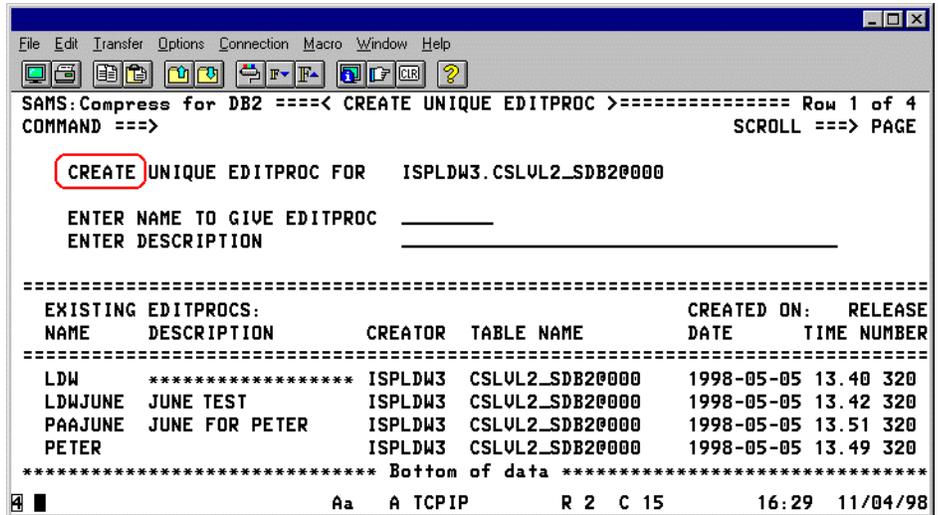
A flexible routine is used in two cases:

- 1 When implementation of the table is excessively time-consuming because of a large number of relationships. Normally, existing relationships have to be re-established when compression is implemented. However, implementing a flexible routine is quicker because you do not have to drop all of the previous relationships.
- 2 When you anticipate that the structure of the DB2 table will change in the future. A flexible routine allows you to add a column to the table after you have implemented compression. A separate batch utility is available to add the columns.

Create Unique EDITPROC Panel

This panel is used to create a unique routine from an optimal compression routine that you want to implement on a table.

FIGURE 11. Unique EDITPROC Panel



To create a unique EDITPROC, move the cursor to the first field on the panel and enter a name to use to identify the EDITPROC. This name must not have previously existed in the designated load library. To assist you in determining whether an EDITPROC name already exists, SAMS:Compress displays a scrollable list of all previously created unique EDITPROCs within the repository. SAMS:Compress validates the name you enter against this list, ensuring that you do not create a duplicate EDITPROC. As further protection, SAMS:Compress links the new EDITPROC to the designated load library without the (R) - replace option on the NAME statement.

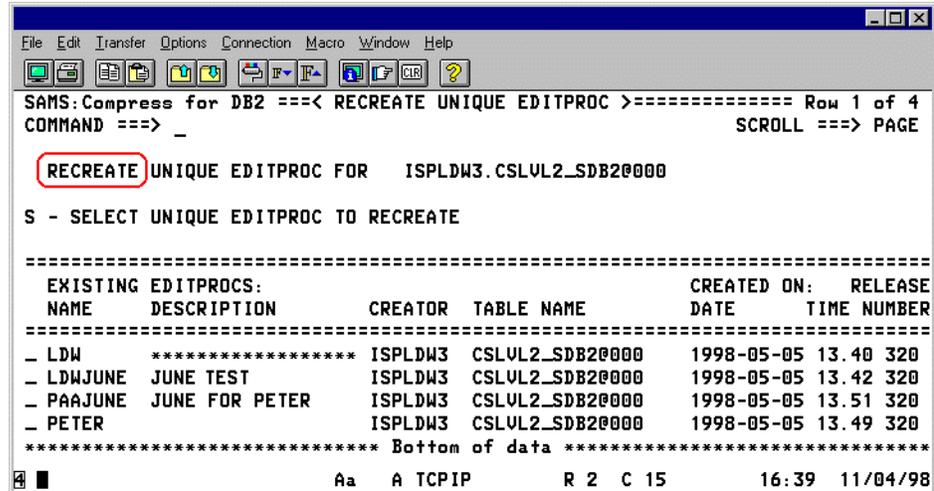
You can optionally enter a description of the unique EDITPROC that is added to the repository, allowing further identification of the EDITPROC.

After you have entered the information, press <ENTER>. SAMS:Compress performs the processing necessary to create the unique EDITPROC. During the create process, the output of the linkage editor is displayed on the screen to let you verify that the process occurred correctly. If a nonzero return code is displayed, refer to your MVS linkage editor manuals to determine why the process was unsuccessful. Correct the problem and repeat the creation process.

Recreate Unique EDITPROC Panel

This panel lets you recreate a unique EDITPROC. It displays the names of all unique EDITPROCs found within the repository for the indicated table and lets you select which one to recreate.

FIGURE 12. Recreate Unique EDITPROC Panel



You can need to recreate a unique EDITPROC for one of the following two reasons.

Note When SAMS:Compress recreates the EDITPROC, it uses the (R) - Replace option of the linkage editor NAME statement. Any existing module in the designated load library with this EDITPROC name is replaced.

- 1 You need to replace a unique EDITPROC that was lost or destroyed. SAMS:Compress protects against this occurrence by storing key values that can be used to recreate the EDITPROC.
- 2 You want to take advantage of enhancements to compression algorithms in new releases of SAMS:Compress. Recreating an EDITPROC incorporates these enhancements into the existing unique EDITPROC.

To recreate a unique EDITPROC, enter an S next to one of the EDITPROCs displayed on the scrollable list of routine names. SAMS:Compress performs the processing necessary to recreate the EDITPROC.

Reported Results Panel

This panel is an information display that contains additional compression analysis information for the selected routine. When you press the <END> key, you return to the Compression Analysis Results Panel.

FIGURE 13. Reported Results Panel

```

SAMS:Compress for DB2 =====< REPORTED RESULTS >=====
COMMAND ==>

                COMPRESSION ANALYSIS RESULTS USING
                COMPRESSION ROUTINE S30XPRES

CREATOR NAME   TABLE NAME      ROWS IN TABLE   ROWS SAMPLED
ISPNEF3        EDU_TABLE          10,004           5,000

          95 PERCENT DATA COMPRESSION EXPERIENCED.
          94 PERCENT SAVINGS IN TABLE SIZE SHOULD BE REALIZED.

-----

                WITHOUT          WITH
                COMPRESSION      COMPRESSION
AUG ROW SIZE   . . . . .      3,680          190
ROWS PER PAGE (MAX 127) . . . . .      1              16
PAGES IN TABLE . . . . .     10,004         630
TABLE SIZE (MBYTES) . . . . .      39              2

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

Aa A TCP/IP      R 2 C 15      17:44 11/04/98

```

The following information is shown on this panel:

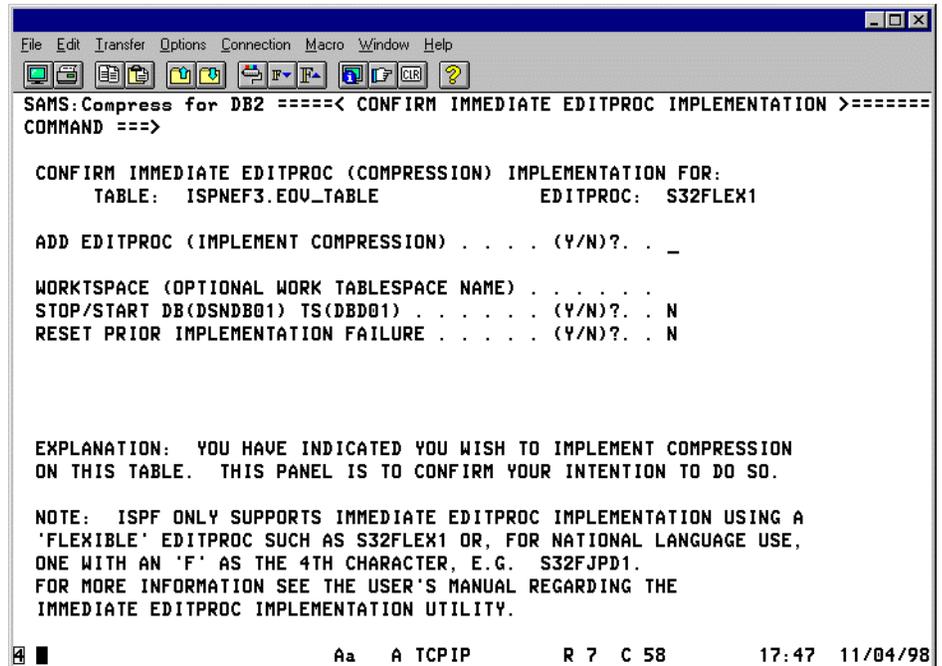
Names	These fields provide the creator and table names.
Rows in Table	This field gives the theoretical number of rows in the table. Note: This number is the product of the number of rows per page times the number of pages in the table.
Rows Sampled	This field gives the number of rows sampled by the selected routine.
Data Compression Experienced	This field presents the row compression percentage, which represents the difference in DB2 data size before and after compression.

Savings in Table Size	<p>This field provides a projection of the savings in tablespace that will be realized when compression is implemented. This value is calculated by using the Number of Rows in Table figure from the Compression Analysis Parameters Panel. Therefore, it is an estimate based on either the information contained in the DB2 catalog or a figure that you supply.</p> <p>DB2 control information, free space, and DB2 page sizes are also used to calculate this value. When tables with fewer than 100 pages of storage size are analyzed, these numbers may seem smaller than they should be because of the extra overhead required by DB2. For example, every DB2 table requires at least two full pages of control information. This control information is never compressed.</p> <p>In addition, this figure reflects the unavoidable DB2 limitation: a maximum of 127 rows per page, regardless of row size.</p>
Average Row Size	<p>This field presents the average row size for the data before and after trial compression. This value indicates how compression has affected overall row size and how DB2 page space will be used. The average row size reflects the length of the data only. It does not include space used by DB2 to manage each row.</p> <p>The average row size (in bytes) is the value SAMS:Compress uses to calculate total space projections. Since no more than 127 rows may be stored in a page of DB2 data storage, the row size value provides valuable information for determining the effectiveness of the compression routines. DB2 pages of storage are usually 4K bytes. If the average row size is such that more than 127 rows could fit on a page, the table is not defined to make efficient use of storage. If a table actually contains 127 rows per page, SAMS:Compress may not be effective.</p>
Rows per Page	<p>This field supplies the average number of DB2 rows that will fit in a page of storage. The maximum number is 127. Use this number to determine how efficiently DB2 makes use of data storage.</p>
Pages in Table	<p>This field shows the size of the DB2 table in pages before and after compression. The value takes into account any free space specifications made for the table when it was created.</p>
Table Size	<p>This field gives the table size in megabytes before and after compression. The value located under “With Compression” can be used to determine the correct allocation of tablespace when implementing compression for this table.</p>

Confirm Immediate EDITPROC Implementation Panel

This panel allows you to confirm the immediate implementation of a flexible routine. The name of the DB2 table to be compressed and the flexible compression routine to be used are shown at the top of the panel.

FIGURE 14. Confirm Implementation Panel



From this panel, you can implement only a flexible routine. Currently, these include the SnnFLEX1 routine and its national language counterparts which are identified by the character F in the fourth position of their name. If you want to implement some other routine, you can use the batch process explained in “Using the Immediate EDITPROC Implementation Utility” on page 55.

When you implement an immediate EDITPROC, SAMS:Compress adds the EDITPROC parameter to the existing table definition. After the EDITPROC is inserted, rows that are inserted or updated are compressed in the table. If an existing row is not updated, it is not compressed. To compress the entire table, perform a DB2 REORG to update all of the table's rows.

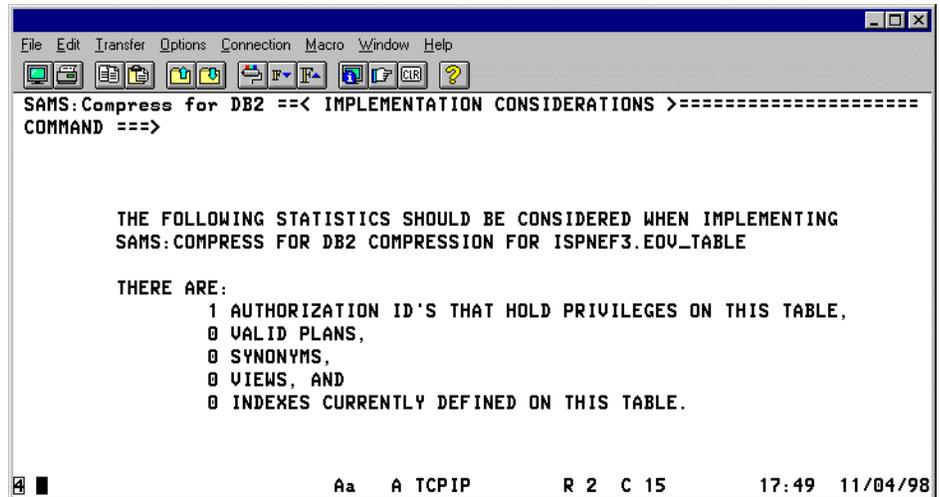
The following parameters are used to confirm the implementation of the routine.

Add EDITPROC	Enter Y to implement the flexible compression routine, or enter N to cancel the operation. You may also press the END PF key to cancel the operation.
Stop/Start DB and TS	Enter Y to stop and start the DB2 directory while inserting the EDITPROC to ensure that updates to the DB2 directory are written both to buffers and disk. This is important only if DB2 should crash and recovery from logs also fails during DB2 restart, leaving the DB2 catalog and directory out of synchronization. Note: This option is not recommended on a production or highly active system.
Reset Prior Implementation Failure	Setting this parameter to Y ensures that the attempt to implement an immediate EDITPROC will not be terminated by a previous failure. If a failure occurs while SAMS:Compress is implementing the EDITPROC, SAMS:Compress can reset some internal failure conditions. With other failures, the table or the SAMS:Compress implementation utility can be left in an unusable state, or processing appears to have completed; the following messages are displayed: <pre>SSA50713T A PRIOR IMPLEMENTATION FAILURE HAS NOT BEEN RESOLVED SSE350F IMMEDIATE EDITPROC IMPLEMENTATION HAS FAILED</pre>
Workspace	This parameter ensures that the SAMS:Compress utility resets and ignores these or other previous failures. Note: The failure state is caused by an ABEND, timeout, or system crash while the utility is executing, not by a SAMS:Compress command or parameter error. Optionally identifies the 1- to 8-character name of a Tablespace in the SAME database, but a DIFFERENT Tablespace as the DB2 Table to be processed. It is used to create a temporary work table required to perform this process, and it is deleted at command completion. The order of tablespace selection is as follows: * WORKTSPACE= (if specified) * IBM DB2 default, which normally goes to Storage Group SYSDEFLT. WORKTSPACE= is only required if the IBM DB2 default does not provide a satisfactory Tablespace. This most likely occurs if your DB2 storage administration policies do not allow the use of SYSDEFLT. If the temporary table cannot be created, the command fails, most likely with a SQL error code 904, "Resource Unavailable...".

Implementation Considerations Panel

This panel shows the number of authorizations, plans, synonyms, views, and indexes related to a DB2 table.

FIGURE 15. Implementation Considerations Panel



To implement compression on an existing table using an optimal routine or any nonflexible standard routine, drop and recreate the table specifying an EDITPROC routine. When you drop a table, DB2 removes all references to the table and drops the authorization, view, and index information. DB2 also invalidates all plans associated with the table. Therefore, when SAMS:Compress is being implemented on an existing table, all previous relationships with DB2 objects should be re-established as needed, after the table is reloaded.

The immediate implementation of a flexible standard routine works around these difficulties.

Batch Compression Analysis

The batch compression analysis facility duplicates the ISPF compression analysis process. The batch facility analyzes a single table at a time. An output report containing the same information that appears on the Reported Results Panels is generated.

FIGURE 16. Batch Compression Analysis JCL

```

File Edit Transfer Options Connection Macro Window Help
-----
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT      ISPDCM1.UTIL.CNTL(ANALYZE) - 01.00      Columns 00001 00072
Command ==>                                     Scroll ==> CSR
000100 //TRIAL      EXEC PGM=IKJEFT01,DYNAMNBR=2,REGION=4096K
000200 //STEPLIB   DD DSN=SHRINK.DB2.LOAD,DISP=SHR
000300 //          DD DSN=DSN220.LOAD,DISP=SHR
000400 //SDB2LOAD  DD DSN=SHRINK.DB2.LOAD,DISP=SHR
000500 //CMDPRINT  DD SYSOUT=*,DCB=(LRECL=133,BLKSIZE=133)
000600 //MSGPRINT  DD SYSOUT=*
000700 //SYSTSPRT  DD SYSOUT=*
000800 //SYSPRINT  DD SYSOUT=*,DCB=(LRECL=133,BLKSIZE=133)
000900 //SYSPRINT  DD SYSOUT=*,DCB=(LRECL=133,BLKSIZE=133)
001000 //SYSUDUMP  DD SYSOUT=*
001100 //TRIAL.SYSIN DD *
001200     ANALYZE CRE=JOHN,TABL=NAMEADR,SKIP=0,MAX=100000,TIME=10,
001300     START=1,LEVEL=3,OPTIMIZE,RPREFIX=SHRINK,RCREATOR=JOHN
001400     ANALYZE CRE=JOHN,TABLE=NAMEQTBL,SKIP=0,MAX=100000,TIME=10,
001500     START=1,LEVEL=3,OPTIMIZE,RPREFIX=SHRINK,RCREATOR=JOHN
001600 //TRIAL.SYSTSIN DD *
001700 DSN SYSTEM(DSN)
001800 RUN PROGRAM(SSA1000) PLAN(SSA0310) -
001900 LIB('SHRINK.DB2.LOAD')
002000 END
-----
Aa A TCP/IP      R 4 C 15      18:01 11/04/98

```

Figure 16 shows the sample JCL used to initiate the batch analysis (member “BATCH” is located in the SAMS:Compress JCL library).

Using the ANALYZE Command

The ANALYZE command shown in Figure 16 specifies which DB2 tables are to be analyzed. The syntax and usage of this command is shown below:

TABLE=	The 1- to 18-character name of the DB2 table to be analyzed.
CREATOR=	The creator associated with the DB2 table to analyze.
RCREATOR=	The creator ID defining the repository name. The repository must exist prior to running batch compression analysis.
RPREFIX=	The prefix ID defining the repository name. The referenced repository must exist.
STARTROW=	The numeric value of the first row in the table to analyze. If this value is not specified, the analysis begins with the first row of the table.
MAXROWS=	Limits the batch analysis to a portion of table data. You can optionally indicate the maximum number of rows to analyze.
ESTROWS=	The estimated (forecasted) number of rows contained in the table.
SKIP=	An optional number of rows to skip between each row analyzed for compression. A value of 9 causes every 10th row of table data to be analyzed.
TIME=	Limits the elapsed time of the batch analysis of each DB2 table. You can optionally indicate a maximum number of minutes for the process.
OPTIMIZE	Indicates that optimal routines will be calculated.

About the Batch Report

Information gained through compression analysis of a table gives an accurate view of how effective compression will be on the data in the table and how compression changes the way DB2 uses tablespace. All statistics generated by the batch utility are placed into the compression analysis repository. The output of batch compression analysis generated is shown below in Figure 17.

FIGURE 17. Batch Compression Analysis Report

```

SDSF OUTPUT DISPLAY ISPLDW3# JOB09759 DSID 106 LINE 119 COLUMNS 02- 133
COMMAND INPUT ==> _
SCROLL ==> PAGE

=====
COMPRESSION ANALYSIS RESULTS USING
COMPRESSION TECHNIQUE (OPTIMAL ALL COLUMNS )

CREATOR NAME TABLE NAME ROWS IN TABLE ROWS SAMPLED
ISPLDW3 CSLUL2_SDB20000 11 11

22 PERCENT DATA COMPRESSION EXPERIENCED
0 PERCENT SAVINGS IN TABLE SIZE SHOULD BE REALIZED

8.8 COMPRESSION PERFORMANCE RELATIVE TO ROUTINE S30SHRVL
7.1 DECOMPRESSION PERFORMANCE RELATIVE TO ROUTINE S30SHRVL

=====
WITHOUT WITH
COMPRESSION COMPRESSION
AUG ROW SIZE . . . . 69 54
ROWS PER PAGE (MAX 127) . . 50 62
PAGES IN TABLE . . . . 3 3
TABLE SIZE (MBYTES) . . . . 0 0
=====
Aa B TCPIP R 4 C 21 17:26 11/05/98

```

The above report shows:

- Names** These fields show the creator and table names.
- Rows in Table** Shows the number of rows in the table.
Note: This value is from the Batch Compression Analysis parameter “ESTROWS.”
- Rows Sampled** Shows the number of rows sampled by the selected routine indicated in the Batch Compression Analysis JCL.
- Data Compression Experienced** Shows the row compression percentage, which represents the difference in DB2 data size before and after compression.

Savings in Table Size

Shows a projection of the savings in tablespace that will be realized when compression is implemented. This value is calculated by using the Number of Rows in Table. Therefore, it is an estimate based on either the information contained in the DB2 catalog or a figure that you supply.

DB2 control information, free space, and DB2 page sizes are also used to calculate this value. When you analyze tables with fewer than 100 pages of storage size, these numbers can seem to small because of the extra overhead required by DB2. For example, every DB2 table requires at least two full pages of control information. This control information is never compressed.

Additionally, this figure reflects the unavoidable DB2 limitation: a maximum of 127 rows per page, regardless of row size.

Relative Performance:

Shows relative CPU usage indicates how well (in terms of CPU requirements) the routine performed against the other routines listed in this report. The routine that used the greatest amount of CPU processing time to compress the sample is assigned a performance value of 10.0. Each of the remaining routine's values should be evaluated in relation to this highest value. For example, if a routine had a value of 5.0, it used half as much CPU processing time as the highest valued routine to achieve its level of compression.

Compression Shows the routine's relative performance when performing compression operations.

Decompression Shows the routine's relative performance when performing decompression operations.

The accuracy of these values can be adversely affected by the operational environment. The amount of time used to analyze a table is affected by the use of cache memory, extended storage, concurrent processing jobs, and PR/SM (LPAR) and VM environments. Each of these factors can cause the recorded simulation time to vary by as much as twenty-five percent.

The following example illustrates how operating in a PR/SM environment can skew the results: SAMS:Compress records the start time at the beginning of a simulation. If the process is interrupted, SAMS:Compress does not include that time in the final analysis. However, in PR/SM, if another LPAR obtains control (an IBM-defined "processor stall"), the active process is stalled in midstream and no indication is passed back to the LPAR. Since SAMS:Compress is not aware of this pause, the stall time is included in the final analysis. This event can potentially result in incorrect figures, depending on how often it occurs during the analysis.

Algorithms are incorporated within the analysis to reduce the potential for inaccurate results, but in some cases the results need to be verified with

multiple analysis runs. To improve the accuracy of the results for a single analysis, you can analyze a large sample of the table (the larger the better) or perform the analysis when activity on the machine is low (consider all LPARs in a PR/SM environment).

Compression Analysis Repository

The SAMS:Compress Compression Analysis Repository contains the results of previous analyses and information used to build and rebuild unique compression routines. SAMS:Compress maintains this information in a repository of SAMS:Compress-created DB2 tables. There are several benefits in having the repository. For instance, you can perform analysis at different times, for different tables, and retain the results for future consideration. And you can view the results using the SAMS:Compress ISPF interface, or develop your own queries using SQL or QMF.

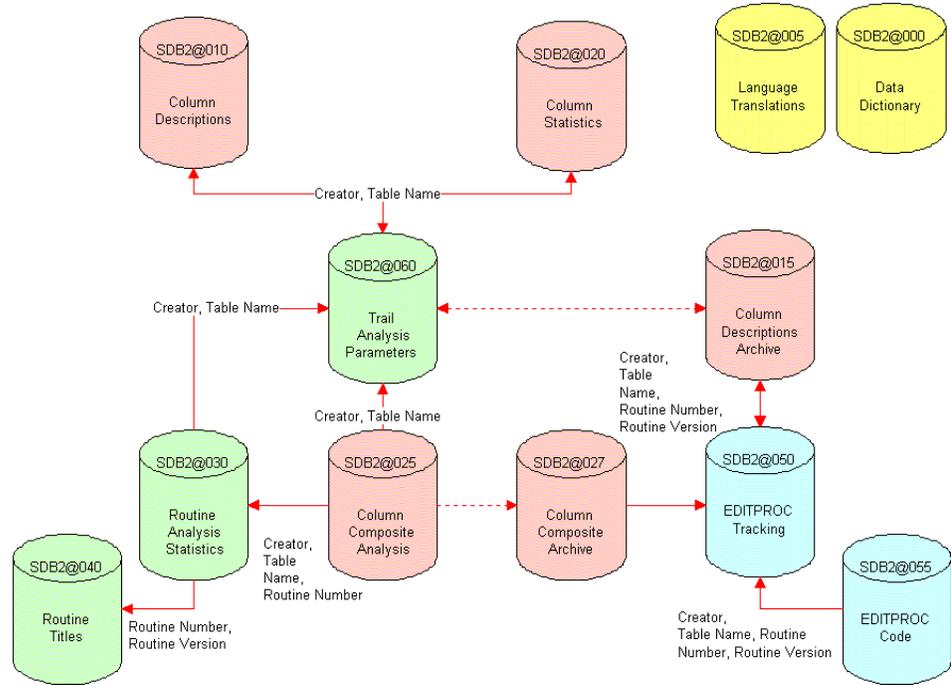
Note Because of the important role the repository plays in the recreate process, only the R (Reset Trial Compression Statistics) function on Figure 5, “Table Selection List and Statistics Panel,” on page 16 to delete table analysis data from the repository.

The repository offers the additional benefit of archiving information that is used in creating optimal routines, allowing you to recreate a routine if necessary. When it creates a unique routine, SAMS:Compress archives the information that is necessary to recreate the EDITPROC (if that should ever be necessary). This feature, coupled with adequate backup procedures, ensures that you always have access to your data when using a unique routine.

Compression Analysis Repository Design

The figure below shows the SAMS:Compress table titles, numbers and linkages. Each table contained within the repository is identified using the creator and prefix names followed by the table number.

FIGURE 18. Repository Table Description and Linkage



The compression analysis repository contains the following tables.

Data Dictionary	Contains the descriptors defining the base tables residing within the compression analysis repository. It is used by SAMS:Compress to create and maintain all of the remaining tables in the repository.
Language Translations	Contains foreign language information for all of the ISPF panels, messages, and prompts. Not yet implemented.
Column Descriptions	Contains a description of the columns for each table analyzed. It is used to determine whether the table has changed since its last analysis.
Column Descriptions Archive	Contains an archived copy of the Column Descriptions table for all DB2 tables that have had unique EDITPROC routines created from them. It is used when recreating a unique EDITPROC compression routine.
Column Statistics	Contains statistics concerning the compression/decompression of each column in a DB2 table that is analyzed. After completed, it is used by COPE to determine the composition of the optimal routines.

Trial Analysis Parameters	Contains the data entered into the Compression Analysis Parameters Panel. It is used to record the parameters of the DB2 tables latest analysis.
Routine Analysis Statistics	Contains the results of the DB2 table's compression analysis. It is used to display the data found on both the Insert/Update and Select perspectives of the Compression Analysis Results Panels.
Column Composite Analysis	Indicates which columns of a DB2 table are used by an optimal routine during compression/decompression. It is used to record the different compositions of the three COPE-created optimal routines.
Column Composite Archive	Contains an archive copy of the rows in the Column Composite Analysis table that corresponds to the optimal routine used to create an EDITPROC. It is used to record the DB2 table columns that are compressed.
Routine Titles	Contains the compression routine descriptions that appear on the Compression Analysis Results Panels.
EDITPROC Tracking	Contains an identifier for each unique EDITPROC created. It includes the DB2 table name on which the EDITPROC is specified, and the SAMS:Compress release number. Note: It is possible to create two unique EDITPROCs with the same name. In this case, the version number is used to distinguish one from the other.
EDITPROC Code	Contains the compiled code for each unique EDITPROC created.

Using the Repository

Because of the important role the repository plays in safeguarding your access to compressed tables, be careful when creating new repositories and when accessing data stored in them.

The following three sections provide guidelines in creating, accessing, and deleting repositories.

Number of Repositories to Create

The optimal number of repositories to create is one. Having only one repository simplifies the decision of which table to implement compression on, since all generated statistics are available. And creating one repository also helps safeguard against deletion of information that is required to maintain and recreate unique EDITPROCs you are using.

Although one repository is preferable, SAMS:Compress lets you make any number of repositories. Perhaps a better setup for your organization is to have two repositories—one containing the test analysis of many tables and the other containing the analysis and unique EDITPROCs for the DB2 tables for which you are implementing compression. A third alternative is to allow any number of test repositories but still maintain only a single production repository. However you select to limit the creation of repositories, remember

that when creating unique EDITPROCs, information describing the EDITPROC must be safeguarded. This safety issue must be addressed before considering the access and desires of the repository users.

Repository Data Access

All data contained within a repository is accessible to you through standard SQL and QMS commands. Access is restrained only by the benefit derived in using the data contained.

The repository tables contain many columns of data that are used internally by SAMS:Compress to analyze your DB2 tables and by the COPE facility to determine the composition of the optimal routines. Others provide information used to create and recreate unique EDITPROCs. Do not delete information from the following tables.

- Column Descriptions Archive
- Column Composite Archive
- EDITPROC Tracking
- EDITPROC Code

Each of the tables listed above contains information that is used to maintain any unique EDITPROCs you may have implemented on one of your DB2 tables.

However, some data contained in the repository can be used to produce meaningful reports. Two tables, Routine Analysis Statistics and EDITPROC Tracking, contain such data. The Routine Analysis Statistics table contains fifteen columns. Many of these columns correspond to the information found on the Compression Analysis Results Panels. The column descriptions are shown below.

Column	Description
S30_CREATOR	Creator name
S30_TABLENAME	DB2 table name
S30_ROUTINE ^a	Compression routine identifier
S30_ROW_CMPR	Percent compression of: row
S30_MEGABYTES	Number of megabytes
S30_RTIMEC	Relative performance: update
S30_RTIMED	Relative performance: select
S30_RTN_NUMBER	Routine number
S30_TBL_CMPR	Percent compression of: table
S30_AVGROW	Average row length
S30_NPAGES	Number of pages
S30_ROWS_PAGE	Rows per page
S30_CMPR_TYPE	Perspective view (B=both; C=Insert/Update; D=Select)

Column (Continued)	Description (Continued)
S30_LEVEL	Reserved for future enhancement
S30_RTN_VERSION	Routine version

- a. An internal SAMS:Compress routine mnemonic. Use this element to extract a representative routine description from the Routine Titles table.

Note The routine description (S40_DESCRIPTION) was extracted from the Routine Titles table (JOHN.Shrink_SDB2@040) using the relationship S30_ROUTINE = S40_NAME.

To demonstrate using SQL on a repository table, the example shown below generates a report of tables that achieve more than fifty percent compression during testing, and are greater than 5 megabytes:

```
SELECT A.S30_CREATOR, A.S30_TABLENAME,
       A.S30_TBL_CMPR, A.S30_MEGABYTES,
       B.S40_DESCRIPTION
FROM JOHN.Shrink_SDB2@030 A, JOHN.Shrink_SDB2@040 B
WHERE A.S30_ROUTINE = B.S40_NAME
AND A.S30_TBL_CMPR >50
AND A.S30_MEGABYTES >5
ORDER BY A.S30_MEGABYTES DESC, A.S30_TABLENAME;
```

The EDITPROC Tracking table contains the following columns.

Column	Description
S50_EDITPROC	Unique EDITPROC name
S50_TABLENAME	DB2 table name
S50_RTN_VERSION	Compression routine version
S50_TIME	Time Unique EDITPROC was created
S50_DISCR	Unique EDITPROC description
S50_CREATOR	Creator name
S50_RTN_NUMBER	Compression routine number
S50_DATE	Date Unique EDITPROC was created
S50_RELEASE_NUM	SAMS:Compress release number

An example SQL accessing this table is shown on the next page.

Deleting Rows and Dropping Tables

Note At no time should you drop only individual tables from a repository.

Only two methods are recommended to remove data from repository tables. The first removes individual compression analysis statistics from within repository tables. The second drops all the tables within a repository.

SAMS:Compress provides a reset function R (Reset Trial Compression Statistics) on the Table Selection List and Compression Statistics Panel (Figure 5 on page 16) to remove unneeded compression analysis results from within a repository. Removing this unneeded data does not affect the operation and recovery capability of existing unique EDITPROCs.

To remove an entire repository you must drop each table.

Note When repository tables are dropped, all unique EDITPROC compression routines created within the repository are lost and cannot be recreated. Therefore, you must ensure that no unique EDITPROC within the repository is currently implemented on a table.

To ensure this, the following SQL statement checks for the existence of any EDITPROCs created by the compression analysis, and are currently implemented on a table:

```
SELECT A.CREATOR, A.NAME, A.EDPROC,
       B.S50_DISCR, B.S50_DATE, B.S50_TIME
FROM SYSIBM.SYSTABLES A,
     JOHN.Shrink_SDB2@050 B
WHERE A.EDPROC = B.S50_EDITPROC
ORDER BY A.CREATOR, A.NAME;
```

Otherwise, you can drop all the tables within the repository. If the above SQL generates a list of EDITPROCs, do not delete the repository.

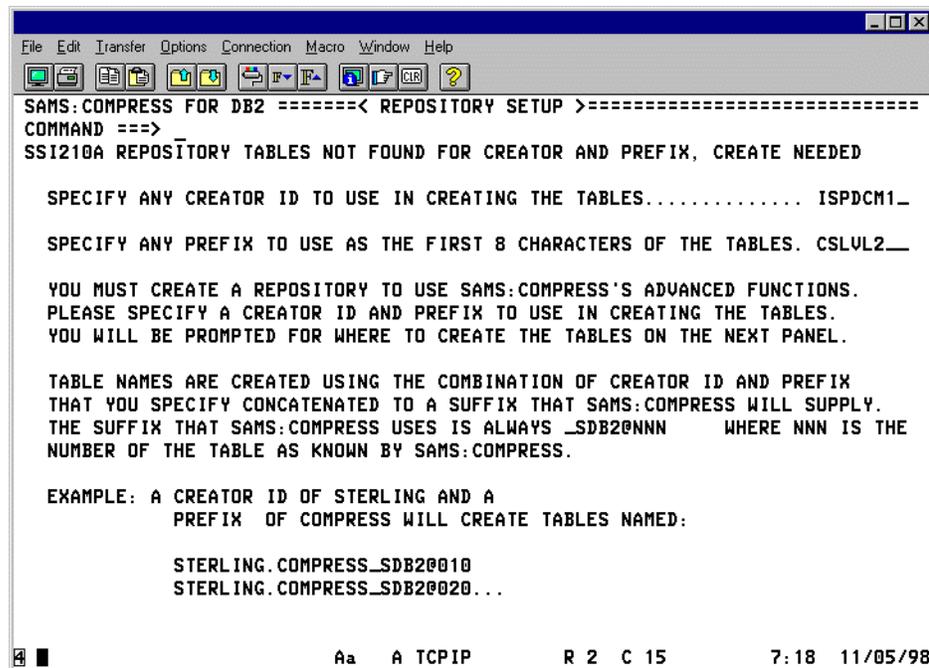
Creating a Repository

Each repository resident in your system is uniquely identified by the creator and prefix names displayed on the Initial Panel. SAMS:Compress provides the two ISPF panels on the following pages to guide you through the setup of a new repository.

Repository Setup Panel

If the repository tables referred to by the combination of creator ID and prefix from the Initial Panel have not been created yet, the Repository Setup panel appears, allowing you to start creating the repository.

FIGURE 19. Repository Setup Panel



Note To perform compression analysis, you must have a repository. You cannot perform an analysis until a repository is created.

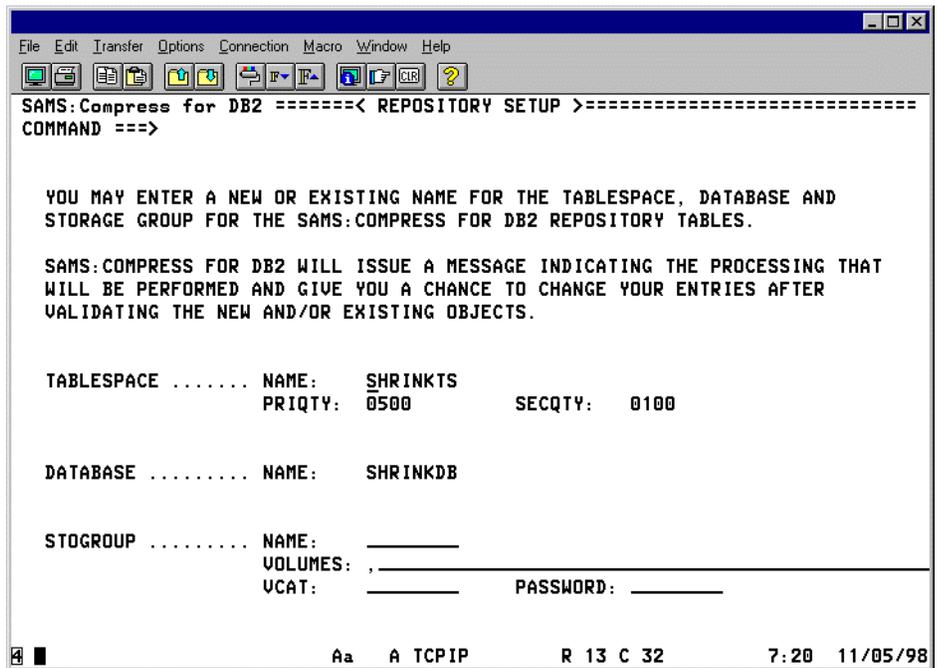
If you do not want to create a new repository, press the <END> key to return to the Initial Panel where you can change the creator ID or prefix values to identify an existing repository.

To create a repository, change the creator ID and prefix values shown on the screen to the labels you want and press the <ENTER> key. The following panel allows you to specify where the tables should be created.

Repository Specification Panel

This panel allows you to specify the location of the new repository tables. Change data by moving the cursor to a field and entering the change.

FIGURE 20. Repository Specification Panel



After all your changes are made, press the <ENTER> key. SAMS:Compress shows a message informing you which items currently exist and which ones need to be created. You can change any field present on the screen if the message does not accurately reflect those items that you want to create. When the objects noted to be created by the current message are correct, press the <ENTER> key. SAMS:Compress creates the repository tables. Information messages are displayed during the create process.

Note Your installation may restrict the ability to create new storage groups or to have access to certain volumes.

After all tables are created, the Repository Setup Panel appears. Press <END> to continue with the compression analysis process.

Implementing Compression

The most important step in using SAMS:Compress is to create a table with the compression routine name and then to compress the table data. SAMS:Compress now allows you to immediately implement compression on a table. Except for the flexible routine, the recommended procedure is to unload, recreate, and reload a table to implement compression. The recommended procedure is shown below in “Implementing a Compression Routine” on page 53. A second procedure, shown in “Using the Immediate EDITPROC Implementation Utility” on page 55, is used to implement flexible routines outside the ISPF panels in batch mode. You can also implement a standard or optimal routine using the batch utility.

Implementing a Compression Routine

To implement compression on an existing table, you must unload, recreate, and reload the table. This unload/recreate/reload process is required by IBM and DB2 for existing tables, but not for new ones, to maintain catalog and data integrity. The process is described below.

- 1 Unload the DB2 table—DB2 users commonly unload and reload tables. IBM provides ISPF panels to assist in this process. Follow normal procedures to unload data. DSNTIAUL is available to unload a table in a format suitable for the reload following the drop/create process.
- 2 Drop the table—The table must be removed from DB2 before it can be recreated with a compression routine. Enter the following SQL command to drop the table:

```
DROP TABLE tablename
```
- 3 Recreate the table—Add the EDITPROC parameter to the existing table definition. Place the compression routine name on the EDITPROC parameter.

Note When the table is dropped, all objects related to the table are also dropped. These need to be recreated later. See step 5

The compression routine to be used must be identifiable to SAMS:Compress by its routine name. The routine name is in one of four categories, described as follows.

- **Standard Routines**—When implementing SnnINTEL, SnnSHRVL, or SnnXPRES, no additional processing is necessary.
- **National Language Compression Routines**—The name of these routines are located on the National Language Compression Panel next to the language ID. No additional processing is necessary prior to using a national language compression routine.
- **Unique EDITPROCs**—Unique EDITPROCs must be created and named by you using the ISPF interface panel before they can be implemented on a table. See “Create Unique EDITPROC Panel” on page 31 for the necessary steps. A unique EDITPROC can only be used to compress the table that it was created for. SAMS:Compress includes checks within the routine to ensure that this requirement is followed. The name you specify in the EDITPROC statement when implementing a unique EDITPROC is the same one you used when the routine was created.

Note If the created routine was placed in a designated load library that is not accessible to DB2, the routine must be moved prior to implementation to a library that is accessible to DB2.

The EDITPROC statement is supplied to SQL to recreate the table. An example of a table definition that has been recreated with the EDITPROC statement is shown below.

FIGURE 21. EDITPROC Statement Example

```
CREATE TABLE SHRINK.COMPRESS
      EMPNO      CHAR(6)      NOT NULL,
      FIRSTNME   VARCHAR(12)  NOT NULL,
      MIDINIT    CHAR(1)      NOT NULL,
      LASTNAME   VARCHAR(15)  NOT NULL
IN SHRNKDB1.SHRNKTS1
EDITPROC S30INTEL;
```

Note You must use Unload/Reload. The use of REORG when adding an EDITPROC is not supported by IBM.

- 4 Reload the table—Use normal IBM procedures to reload the table.
- 5 Re-establish all objects related to the table—Re-establish authorization requests, plans, views, and other objects related to the table before it was dropped. Follow normal procedures to perform this step.

After all these procedures have been performed, you simply use the tables. DB2 compression is transparent to users of the data.

Immediate EDITPROC Implementation

Note You can use this utility to implement any of the standard or optional compression routines. However, you must unload and reload the table when implementing any of these routines.

The SAMS:Compress Immediate EDITPROC Implementation Utility lets you dynamically add, change, or remove the EDITPROC associated with a DB2 table. This allows compression to be implemented without having to unload, drop, create, and reload the table.

This utility assumes that you are implementing a flexible routine, which is designed to allow for this capability. The flexible routine is the only routine that allows DB2 to process a table containing a mixture of compressed and uncompressed rows. The routine also allows you to later add columns to a compressed table.

Using the Immediate EDITPROC Implementation Utility

When you use the utility, DB2 must be active as the utility performs select and update operations against the DB2 Catalog. The utility operates as a batch job using the TSO Terminal Monitor Program (TMP). Sample JCL to run the utility is included in the SAMS:Compress JCL library in member EDITUTIL.

- 1 Before using EDITUTIL, make the following modifications.
- 2 Update the JOBCARD to your installation standards.
- 3 Update the STEPLIB DD to point to your installation's SAMS:Compress and IBM DB2 Load libraries.
- 4 Update the SYSLIB DD to point to your installation's SAMS:Compress Load libraries.
- 5 Update EDITPROC.SYSTSIN to include the DB2 subsystem name in the DSN command with the name of the DB2 subsystem within which the database resides (for example, DSN SYSTEM(DB2S)).
- 6 Insert the commands and parameters available from this utility in EDITPROC.SYSIN as necessary.

Utility Commands

If the table you are going to implement compression on contains data, see the section "Preserving Your Data" on page 57 before running this utility. The commands available with this utility include:

SIMULATE - Indicates whether all the following commands are to be simulated. Parameters include:

```
ON          (Default value)
OFF
```

ON indicates that no updates will occur but all commands are validated for syntax and information messages are issued. OFF disables the simulation.

ADDEDIT - Adds an EDITPROC to a DB2 table. Parameters include:

```
TABLE=          CREATOR=      EDITPROC=
NOEDITCHECK    SSDB01         SIMULATE
WILLRELOAD     WORKTS=
```

CHANGEEDIT - Changes the EDITPROC specified for a table from one compression routine to another. Parameters include:

TABLE=	CREATOR=	EDITPROC=
NOEDITCHECK	SSDB01	SIMULATE
WILLRELOAD	WORKTS=	

Note Each of the parameters is fully described below.

REMOVEEDIT - This command removes an EDITPROC from a DB2 table. Parameters include:

TABLE=	CREATOR=	EDITPROC=
NOEDITCHECK	SSDB01	SIMULATE
WILLRELOAD	WORKTS=	

TABLE= - Identifies the 1- to 18-character name of the DB2 table to be processed.

CREATOR= - Identifies the 1- to 8-character name of the creator of the DB2 table to be processed.

EDITPROC= - Identifies the 1- to 8-character name of the new EDITPROC name to be added to the DB2 table definition. You must identify the complete name of a SAMS:Compress compression routine.

The SAMS:Compress flexible routine (SnnFLEX1) or its National Language counterparts can be inserted into an existing, noncompressed table that already contains data. It also allows for the addition of columns in the future.

NOEDITCHECK - Causes SAMS:Compress to bypass verifying that the EDITPROC name specified is a recognized SAMS:Compress compression routine. You must specify this parameter when the EDITPROC is an optimal routine that you created.

SIMULATE - Causes simulated execution of the individual command. When used, the utility validates the parameters and produces the output on the SYSPRINT dataset, but does not implement the EDITPROC. Optional parameter.

Note This option is not recommended on a production or highly active system.

SSDB01 - Causes updates to the DB2 directory, created while the EDITPROC is being added, to be written immediately to disk, by stopping and starting the DB2 directory. This is important only if DB2 crashes and recovery from logs also fails during DB2 restart, leaving the DB2 catalog and directory out of synchronization.

WILLRELOAD - Identifies that you have unloaded the DB2 table before executing this utility and reloads it afterwards. This parameter is necessary only when you are using a nonflexible routine (that is, a standard or optimal routine). Nonflexible routines cannot be used on tables that simultaneously contain compressed and uncompressed rows.

Use the following steps whenever you are specifying a nonflexible routine in this utility:

- Unload the table.
- Execute the utility using the WILLRELOAD parameter with the ADDEDIT, CHANGEEDIT, or REMOVEEDIT commands.
- Reload the table.

WORKTSPACE= - Optionally identifies the 1- to 8-character name of a Tablespace in the SAME Database, but a DIFFERENT Tablespace as the DB2 Table to be processed. It is used to create a temporary work table required to perform this process, and it is deleted at command completion.

The order of tablespace selection is as follows:

- WORKTSPACE= (if specified)
- IBM DB2 default, which normally goes to Storage Group SYSDEFLT.

WORKTSPACE= is only required if the IBM DB2 default does not provide a satisfactory Tablespace. This occurs if your DB2 storage administration policies do not allow the use of SYSDEFLT.

If the temporary table cannot be created, the command fails, most likely with a SQL error code 904, "Resource Unavailable..."

Preserving Your Data

The flexible compression routine allows DB2 tables to simultaneously contain both compressed and uncompressed rows, and allows columns to be added after the table is loaded. You may use this routine or its National Language counterparts on non-empty, uncompressed tables. The other compression routines do not incorporate these capabilities.

The flexible National Language compression routines are identified by an F placed in the fourth position of their names (for example, S32FJPD1).

If you change or remove the EDITPROC from a table, the table must be unloaded and then reloaded. This allows SAMS:Compress to uncompress the entire table using the old EDITPROC. Do not change or remove an EDITPROC unless you have done an UNLOAD or a REORG with UNLOAD ONLY (decompressing the data) for the table or the table is empty (or new). After changing or removing the EDITPROC, you must LOAD the table, which causes the data to be compressed by the new routine.

Restrictions on the Immediate EDITPROC Implementation Utility

The utility contains the following restrictions and limitations:

- Security/Authorization requirements mandate that you have one of the following authorization levels:
 - SYSADM
 - SYSCTRL
 - DBADM and you must be the recipient of a “GRANT UPDATE (EDPROC , RECLENGTH) ON TABLE SYSIBM.SYSTABLES TO your user ID”.
- This utility supports only certain compression routines.

You can enter as many commands as you want in the input stream. If an error is encountered, command processing is terminated. Subsequent commands are checked for syntax errors and not executed (as if in simulate mode).

Using Additional Programs on Compressed Tables

SAMS:Compress allows you to use utility programs to perform the following functions:

- Add columns to a compressed table
- Access compressed data within DB2 tables without using DB2
- Allow Third Party Programs to access DB2 tables that are compressed by SAMS:Compress

Add Column Utility

Note You can use this utility to add columns to tables using any of the standard or optional compression routines. However, you must unload and reload table when adding a column.

The Add Column Utility provides the capability to immediately add a column to a table, even if the table currently has an EDITPROC, without having to unload, drop, create, and reload the table.

This utility assumes that you have implemented a flexible routine, which is designed to allow for this capability. The flexible routine is the only routine that allows you to later add columns to a compressed table.

Using the Add Column Utility

When you use the utility, DB2 must be active as the utility performs select and update operations against the DB2 Catalog. The utility operates as a batch job using the TSO Terminal Monitor Program (TMP). Sample JCL to run the utility is included in the SAMS:Compress JCL library in member EDITUTIL.

Before using EDITUTIL, make the following modifications:

- 1 Update the JOBCARD to your installation standards.
- 2 Update the STEPLIB DD to point to your installation's SAMS:Compress and IBM DB2 Load libraries.

- 3 Update the SYSLIB DD to point to your installation's SAMS:Compress Load libraries.
- 4 Update EDITPROC.SYSTSIN to include the DB2 subsystem name in the DSN command with the name of the DB2 subsystem within which the database resides, for example, DSN SYSTEM(DB2S).
- 5 Insert the commands and parameters available from this utility in EDITPROC.SYSIN as necessary.

Utility Commands

If the table to which you are going to add a column currently contains data, see “Preserving Your Data” on page 62 before running this utility. The commands available with this utility include:

SIMULATE

Indicates whether all of the following commands are to be simulated.

ON - Indicates that no updates occur but all commands are validated for syntax and information messages are issued. This is the default.

OFF - Disables the simulation.

ADDCOLUMN

Describes the column to be added to the table. Parameters available on the command include:

TABLE= - Identifies the 1- to 18-character name of the DB2 table to be processed.

CREATOR= - Identifies the 1- to 8-character name of the creator of the DB2 table to be processed.

COLNAME= - Identifies the 1- to 18- character name of the column to be added to the table.

TYPE= - Identifies the DB2 column type to be added. The table shown for the **LENGTH=** parameter lists the column types available.

LENGTH= - Indicates the 1- to 5-numeric digits specifying the length or size for the type of column. Specify the value as nnnnn, (nnnnn), or (nnnnn,nnnnn). The following table shows deviations between standard DB2 column definitions and those available when using this utility.

Note LONG-type columns are created without LONG and with proper length calculated, allowing 10 bytes for an EDITPROC.

Column Type	Length/Size	Byte Count ^a
INTEGER	Error if specified	No change
SMALLINT	Error if specified	No change
FLOAT	Optional	No change
REAL	Error if specified	No change
DOUBLEPRECISION	Error if specified	No change
DECIMAL	Optional	No change

Column Type	Length/Size	Byte Count ^a
NUMERIC	Optional	No change
CHARACTER ^b	1-254 ^c	LEN
VARCHAR ^b	1-32767 ^c	LEN + 2
LONGVARCHAR	Error if specified	Calculated by Utility
GRAPHIC ^b	1-127 ^c	2 * LEN
VARGRAPHIC ^b	1-32767 ^c	2 * LEN + 2
LONGVARGRAPHIC	Error if specified	Calculated by Utility
DATE	Error if specified	No change
TIME	Error if specified	No change
TIMESTAMP	Error if specified	No change

a. Add one additional byte to the Byte Count shown if nulls are allowed (NOTNULL=N)

b. Additional information is provided in the DB2 SQL Reference

c. You must specify a Length value

NOTNULL= - Indicates whether this column should be 'NOT NULL WITH DEFAULT'. Specify Y to not allow or N (the default value) to allow the null value.

FOR= - Indicates the subtype of a character string column, and will be used in a 'FOR subtype DATA' clause. This parameter is optional with no default value. Values include SBCS, MIXED, or BIT.

FIELDPROC= - Identifies the 1- to 8- character name of the field procedure exit name. This parameter is optional with no default value.

FIELDPARAM= - Allows you to specify unique parameter(s) that are required. Enter the parameters using single quotation marks and parentheses as shown below, using no more than a total of 255 characters.

```
'(parameter,parameter,...)'
```

NOEDITCHECK - Causes SAMS:Compress to bypass verifying that the EDITPROC name in use is a recognized SAMS:Compress compression routine. If you are using a unique EDITPROC, a new one must be created after the column is added.

SIMULATE - Causes simulated execution of the individual command. When used, the utility validates the parameters and produces the output on the SYSPRINT dataset. This parameter is optional.

SSDB01 - Causes updates to the DB2 directory, created while the EDITPROC is being added, to be written to both buffers and to disk, by stopping and starting the DB2 directory. This is important only if DB2 crashes during DB2 restart and you are unable to recover from logs, leaving the DB2 catalog and directory out of synchronization.

WILLRELOAD - Identifies that you have unloaded the DB2 table before executing this utility and will reload it afterwards or that the table is empty.

Note This parameter is optional; it can only be specified with FIELDPROC=, and cannot be specified with NOTNULL=Y.

Note This option is not recommended on a production or highly active system.

This parameter is necessary only when you are using a nonflexible routine (that is, a standard or optimal routine). Nonflexible routines cannot be used on tables that simultaneously contain compressed and uncompressed rows.

Use the following steps whenever you are specifying a nonflexible routine in this utility:

- Unload the table.
- Execute the utility using the WILLRELOAD parameter.
- Reload the table.

WORKTSPACE= - Optionally identifies the 1- to 8- character name of a Tablespace in the SAME Database, but a DIFFERENT Tablespace as the DB2 Table to be processed. It is used to create a temporary work table required to perform this process, and is deleted at command completion.

The order of tablespace selection is as follows:

- WORKTSPACE= (if specified)
- IBM DB2 default, which normally goes to Storage Group SYSDEFLT.

WORKTSPACE= is only required if the IBM DB2 default does not provide a satisfactory Tablespace. This occurs if your DB2 storage administration policies do not allow the use of SYSDEFLT.

If the temporary table can not be created, the command fails, most likely with a SQL error code 904, "Resource Unavailable..."

Preserving Your Data

The flexible compression routine allows DB2 tables to simultaneously contain both compressed and uncompressed rows, and allows columns to be added after the table is loaded. You can use this routine or its National Language counterparts on non-empty, uncompressed tables. The other compression routines do not incorporate these capabilities and require you to take the steps outlined under the WILLRELOAD option to preserve your data.

The flexible National Language compression routines are identified by an F placed in the fourth position of their name (for example, S32FJPD1).

Restrictions on the Add Column Utility

The utility contains the following restrictions and limitations.

- Security/Authorization requirements require that you have one of the following authorization levels.
 - SYSADM
 - SYSCTRL
 - DBADM and you must be the recipient of a "GRANT UPDATE (EDPROC , RECLENGTH) ON TABLE SYSIBM.SYSTABLES TO your user ID".
- This utility supports only certain compression routines.

You can enter as many commands as you want in the input stream. If an error is encountered, processing is terminated. Subsequent commands are checked for syntax errors and not executed as if in simulate mode.

Print Utility

The Print Utility prints pages of a DB2 table directly from disk. This utility produces output similar to the IBM DSN1PRNT Utility and functions in the same manner. It prints 4K pages from compressed and uncompressed tables. 32K page printing is not currently supported. When you print pages from a compressed table, the utility automatically decompresses the rows within the pages using the correct EDITPROC compression routine, regardless of whether the EDITPROC is a standard or optimal compression routine.

Using the Print Utility

When you use the Print Utility, DB2 must be active to ensure that the correct EDITPROC is used and to obtain other table information that is needed for the decompression. The Print Utility operates as a batch job using a TSO batch program. You can find model JCL to run the Print Utility in the SAMS:Compress JCL library, named PRNTUTIL.

To run the utility you must first modify the model JCL as follows.

- 1 Update the JOBCARD to your installation standards.
- 2 Update the SYSLIB DD to point to your installation's DB2 LOAD library.
- 3 Update the SYSUT1 DD to point to the DB2 tablespace from which you wish to print the pages. If the tablespace is partitioned, pay special attention to the partition number in the dataset name.
- 4 Update the SYSTEM parameter to contain the name of the DB2 subsystem within which the table is defined.
- 5 Update the following parameters to control the Print Utility processing. (The TABLE= and CREATOR= parameters are required. All others are optional.)

TABLE=	Table name (1-18 character string) associated with the DB2 table to be printed.
CREATOR=	Creator associated with the DB2 table to be printed.
SPAGE=	Numeric value of the first page in the table to print. Starting with 0 as the first page of the table. The first two pages in the tablespace are not printed since they do not contain data.
EPAGE=	Numeric value of the last page in the table to print.
SVALUE=	Character string to scan for within a page that will trigger a print of that page. You can enter from 1 to 20 characters using the following format: SVALUE='(xxxxxxx)' where xxxxxxx is a character string

XVALUE= String of hex digits to scan for within a page that will trigger a print of that page. You can enter from 1 to 40 hex digits using the following format:

XVALUE='(xxxxxxx)' where xxxxxxx is a string of paired hex digits

CHECK Specify that a compression and decompression validity check be performed of the entire tablespace. This diagnostic parameter should be used only under the direction of the SAMS:Compress Technical Support Center.

CONTINUE Specify that the CHECK command continue to scan the rest of the tablespace after any errors are encountered.

- 6 Submit the job after you have completed the updates.

The utility validates the control parameters and reports any errors in the CMDPRINT dataset. If no errors are found, the utility produces the page print output on the SYSPRINT dataset.

The following two examples demonstrate the use of the JCL parameters for the Print Utility.

- 1 This example scans pages 3 through 6 for a character value of "MONTANA" and prints the entire page if it is found within the page.

```
TABLE=PAYROL,CREATOR=USER1,SPAGE=3,EPAGE=6,SVALUE='(MONTANA)'
```

- 2 This example performs a check of the entire table to ensure that every row is correctly compressed and decompressed. It continues the scan if an error is encountered, after printing diagnostic information about the error.

```
TABLE=PERSON,CREATOR=USER2,CHECK,CONTINUE
```

Restrictions on the Print Utility

The SAMS:Compress Print Utility contains the following restrictions and limitations.

- 32K pages cannot be printed with this utility.
- Only one table can exist in the tablespace that is printed.
- Only one option is allowed for printing (FORMAT) since the printing of uncompressed data can be performed using the DB2 DSN1PRNT Utility.

Third Party Program Capability

Because many third party programs access data from DB2 tables without relying on the DB2 system, SAMS:Compress provides a Third Party Program Capability that allows these programs to access compressed data within the DB2 tables. SAMS:Compress compression routines operate as EDITPROCs within DB2. To access compressed data, this utility provides copies of the standard compression routines that do not rely on DB2.

The SAMS:Compress Third Party Program Capability works only on tables that are compressed using a standard routine. A procedure is available that allows access to tables that are compressed using optimal routines. To obtain this procedure, contact the SAMS:Compress Technical Support Center.

To access the modified compression routines, revise your third party program job stream by changing the STEPLIB DD to point first to the SHRINK.DB2.BATCH.LOAD library and then to your DB2 LOAD library.

As an alternative to modifying your job streams, you can copy the compression routines from SHRINK.DB2.BATCH.LOAD to the library containing the third party programs. We encourage the STEPLIB approach due to maintenance considerations. Installation steps 3 and 4 describe the compression routine copy process.

Installing

This section contains the procedure for installing SAMS:Compress. It includes the following information:

- Installation Tape Contents
- Installation Check List
- Installation Steps

Installation Tape Contents

The SAMS:Compress installation tape contains the following nine files.

- 1 DSN=SHRINK.DB2.JCL—Contains installation JCL, control statements, and sample JCL to run the batch utilities.
- 2 DSN=SHRINK.DB2.LOAD—Contains compression exits and compression analysis routines.
- 3 DSN=SHRINK.DB2.BATCH.LOAD—Contains the Print and Third Party Program Companion Utilities.
- 4 DSN=SHRINK.DB2.CLIST—Contains the CLIST to start the ISPF compression analysis function.
- 5 DSN=SHRINK.DB2.DBRMLIB—Contains database request modules for compression analysis.
- 6 DSN=SHRINK.DB2.ISPMLIB—Contains messages for ISPF compression analysis.
- 7 DSN=SHRINK.DB2.ISPPLIB—Contains panels for ISPF compression analysis.
- 8 DSN=SHRINK.DB2.ISPSLIB—Contains skeletons for building the repository for the ISPF compression analysis.
- 9 DSN=SHRINK.DB2.ISPTLIB—Contains temporary data for ISPF compression analysis.

Installation Checklist

The checklist is a summary of the steps required to install SAMS:Compress. It can be used as a planning tool. For installation details, see “Installation Steps” on page 68.

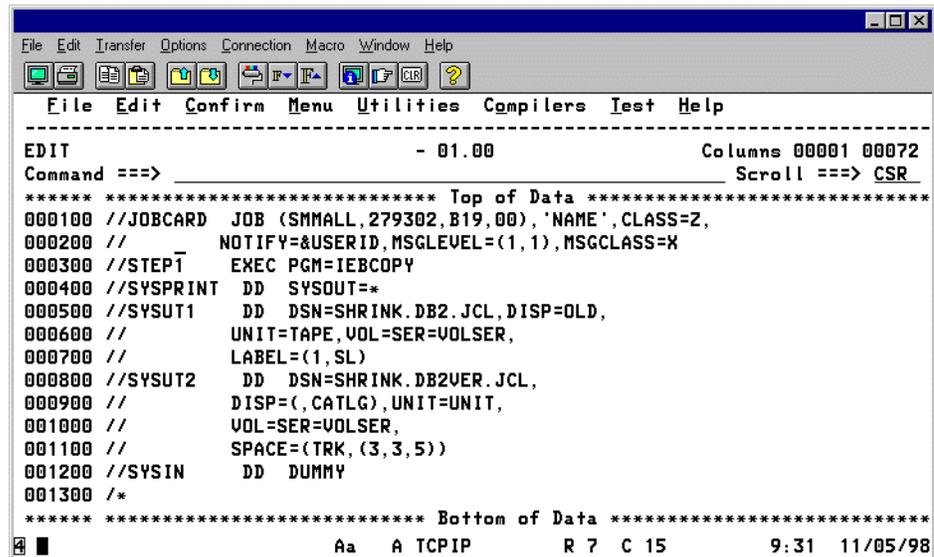
- Unload the distribution libraries.
- Modify the JCL for the environment.
- Run the modified JCL.
- Copy compression exits to the DB2 exit library or an authorized DB2 load library.
- Modify and move the CLIST to run compression analysis to a TSO CLIST library.
- Install the ISPF libraries for SAMS:Compress ISPF processing.
- Bind a plan for analysis application and authorize users to access the plan.
- Bind a plan for the Print Utility and authorize users to access the plan.
- Bind a plan for the Immediate EDITPROC Implementation Utility and authorize users to access the plan.
- Run the Installation Verification Procedure.

Installation Steps

To install SAMS:Compress, follows the steps below.

- 1 Copy the first file on the distribution tape to SHRINK.DB2.JCL. The PDS contains control statements and JCL used in subsequent steps of the installation process. Use the JCL below as an example when making the copy.

FIGURE 22. JCL to Copy the JCL PDS from Tape to Disk



```

EDIT                                     - 01.00                                Columns 00001 00072
Command ==>                               Scroll ==> CSR
***** ***** Top of Data *****
000100 //JOB CARD JOB (SMALL,279302,B19,00),'NAME',CLASS=Z,
000200 //          NOTIFY=&USERID,MSGLEVEL=(1,1),MSGCLASS=X
000300 //STEP1    EXEC PGM=IEBCOPY
000400 //SYSPRINT DD SYSOUT=*
000500 //SYSUT1   DD DSN=SHRINK.DB2.JCL,DISP=OLD,
000600 //          UNIT=TAPE,VOL=SER=VOLSER,
000700 //          LABEL=(1,SL)
000800 //SYSUT2   DD DSN=SHRINK.DB2VER.JCL,
000900 //          DISP=(,CATLG),UNIT=UNIT,
001000 //          VOL=SER=VOLSER,
001100 //          SPACE=(TRK,(3,3,5))
001200 //SYSIN   DD DUMMY
001300 /*
***** ***** Bottom of Data *****
Aa  A TCP/IP          R 7 C 15          9:31 11/05/98

```

- 2 The JCL PDS contains member INSTALL. Use the INSTALL member to unload the remainder of the distribution files from the tape to disk.

Modify the JCL with the correct dataset names, unit names, SAMS:Compress version number, and volume serials for the installation. Run it to create the remaining SAMS:Compress distribution libraries. The volser is printed on the label of the installation tape.

- 3 To prepare the SAMS:Compress compression exits, copy the following SAMS:Compress compression exit routines from SHRINK.DB2.LOAD into any authorized library connected to DB2:

- SnnINTEL
- SnnSHRVL
- SnnXPRES
- SnnFLEX1
- SnnVALID & SnnIMPL—These are required only if you want to perform immediate EDITPROC implementation. They do not have BATCH.LOAD equivalents.

- 4 To incorporate national language compression, copy any of the following compression exits from SHRINK.DB2.LOAD to the same library used in the step above.

Note The letters nn are used in the routine names to represent the release number of the SAMS:Compress product in which the routine was introduced.

Note IBM recommends copying compression exits to the DSNEEXIT library initially defined by IBM for each DB2 subsystem.

National Language	Compression Exit (Routine Name)
American	SnnINTEL ^a
English (U.K.)	SnntENsn
French	SnntFRsn
German	SnntGESn
Italian	SnntITsn
Japanese DBCS (Kanji)	SnntJPDn
Japanese SBCS (Katakana)	SnntJPSn
Korean DBCS (Kanji)	SnntKODn
Korean SBCS (Katakana)	SnntKOSn
Portuguese	SnntPOsn
Spanish	SnntSPsn

a. The intelligent compression routine is used for the American national language.

The naming convention for the national language compression routines is: **Snntccsn**

- S** Shrink
- nn** Version
- t** Compression Routine Type:
 - I Intelligent
 - F Flexible National Language Counterparts
- cc** Country The valid identifiers are shown in the table above.

- s Character set:
 S Single Byte
 D Double Byte
 - n Routine number. The routine number allows for different Huffman compression tables to exist for a country or character set usage.
- 5 The CLIST library, SHRINK.DB2.CLIST, contains the CLIST member SDB2 used to start the compression analysis facility from the ISPF TSO COMMAND screen. Install the CLIST into any TSO CLIST library accessible to those individuals who use the SAMS:Compress compression analysis facilities. The CLIST is divided into five sections. The first section contains a list of dataset parameters. Modify these parameters to conform to your specific installation. The remaining sections of the CLIST require that no changes be made to them.

Note Only implemented languages have routines in the library.

Below is an example of the first section of the CLIST member SDB2. Examine this portion of the member and verify that the parameters listed below are correct for the specified installation.

FIGURE 23. CLIST Member SDB2

```

File Edit Transfer Options Connection Macro Window Help
-----
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT (SDB2) - 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR
000100 PROC 0 HLQ(SHRINK) VER(320)
000200 /*****
000300 /*
000400 /* ***** CHANGE THE DATASET NAMES BELOW FOR YOUR INSTALLATION NAMES *
000500 /* PLEASE BE SURE THAT COMMENTS DO NOT EXTEND INTO COL 72 *
000600 /*
000700 /*****
000900 /* NOTE - DELETE THE STEPLIB COMMAND STATEMENT IF DB2 LOAD LIBRARY *
001000 /* IS ALREADY ALLOCATED TO YOUR SESSION. *
001100 /*****
001200 SET &DB2LIB = &STR(DSN220.LOADLIB) /* DB2 LOADLIB 22
001300 SET &FCLIB = &STR(&HLQ..DB2&VER..CLIST) /* CLIST LIBRARY
001400 SET &FPLIB = &STR(&HLQ..DB2&VER..ISPPLIB) /* PANEL LIBRARY
001500 SET &FMLIB = &STR(&HLQ..DB2&VER..ISPLIB) /* MESSAGE LIBRARY
001600 SET &FSLIB = &STR(&HLQ..DB2&VER..ISPLIB) /* SKELETON LIBRARY
001700 SET &FTLIB = &STR(&HLQ..DB2&VER..ISPTLIB) /* TEMPORARY LIBRARY
001800 SET &FELIB = &STR(DSN220.DSNEKIT) /* DB2 EDITPROC LOADLIB
001900 SET &FOLIB = &STR(&HLQ..DB2&VER..BATCH.LOAD) /* BATCH LOAD LIBRARY
002000 SET &FOLIB = &STR(&HLQ..DB2&VER..LOAD) /* LOAD LIBRARY
002100 SET &UNIT = &STR(SYSDA) /* ESOTERIC UNIT
-----
Aa A TCP/IP R 4 C 15 10:13 11/05/98
  
```

Modify the PROC statement setting the HLQ variable to the high level qualifier of your SAMS:Compress libraries, and the VER (version) variable to the suffix of the second qualifier.

- &DB2LIB Specifies the IBM DB2 load library that contains the DSN program.
- &FCLIB Specifies the SAMS:Compress CLIST library containing the command SSE0310. See instructions at the end of this step to determine whether you should comment out the line.

&FPLIB	Specifies the ISPF panel library. You can change the name to conform to your naming conventions.
&FMLIB	Specifies the message library. You can change the name to conform to your naming conventions.
&FSLIB	Specifies the skeleton library. You can change the name to conform to your naming conventions.
&FTLIB	Specifies the temporary library. You can change the name to conform to your naming conventions.
&FELIB	Specifies the EDITPROC link library. You can change the name to conform to your naming conventions. Unique optimal EDITPROCs are link-edited to this library, if you create any.
&FOLIB	Specifies the load library. You can change the name to conform to your naming conventions.
&FBLIB	Specifies the batch load library. You can change the name to conform to your naming conventions.
&UNIT	Specifies the device type on which to allocate the work files.

The CLIST member SSE0310, contained within the CLIST library (SHRINK.DB2.CLIST), must be accessible to TSO when you are running the ISPF compression analysis facilities. Follow the steps below to make member SSE0310 accessible to TSO.

- If you are running TSO/E Version 2.1 or higher, you do not need to make any changes. The CLIST uses the ALTLIB command to access additional CLIST libraries. Otherwise, comment the ALTLIB commands shown below out of the member and continue with the following alternatives.

```
ALTLIB ACTIVATE APPLICATION(CLIST) DS('&FCLIB') QUIET
      STEPLIB ('&DB2LIB')
ALTLIB DEACTIVATE APPLICATION(CLIST) QUIET
```

- Modify your SYSPROC concatenations to include the SHRINK.DB2.CLIST library.
- Move SSE0310 to a library that is already in your SYSPROC concatenations.
- Uncomment the following three lines from the startup CLIST:

```
/* ALLOC F(SYSPROC) DA('SYS1.CMDPROC)
/* ALLOC F(SYSPRO1) DA('&FCLIB')
/* CONCAT (SYSPROC,SYSPRO1)
```

- 6 The ISPMLIB, ISPPLIB, ISPSLIB, and ISPTLIB distribution libraries contain members that are needed for ISPF compression analysis. Ensure that these libraries are available to your ISPF session.

Note This option can be used only if you have the TSO CONCAT command available. CONCAT is a shareware type of command.

- 7 SAMS:Compress provides a member in the SAMS:Compress INSTALL distribution library containing JCL to BIND each PLAN needed by SAMS:Compress. The member BIND is used for the SAMS:Compress ISPF compression analysis, Print Utility and Immediate EDITPROC Implementation Utility.

FIGURE 24. JCL Used to Bind the Plan (BIND)

```

EDIT (BIND) - 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data *****
000001 //BIND EXEC PGM=IKJEFT01,DYNAMNBR=2,REGION=2048K
000002 //STEPLIB DD DSN=DSN220.DSNLOAD,DISP=SHR
000003 //DBRMLIB DD DISP=SHR,DSN=SHRINK.DB2.DBRMLIB
----- 4 Line(s) not Displayed -----
000008 //SYSTSIN DD *
000009 DSN SYSTEM(DSN)
000010 BIND PLAN(SSA0310) MEMBER(SSA0SQL -
000011 SSA0212 SSA0214 SSA0216 SSA0218 SSA0310 SSA0312 SSA0314 SSA0330 -
000012 SSA0340 SSA0350 SSA0355 SSA0356 SSA0357 SSA0370 SSA0700 SSA0710 -
000013 SSA1000 SSA2000 SSA2300 SSA2310 SSA2315 SSA2317 SSA2320 SSA2330 -
000014 SSA2340 SSA2360 SSA2370 SSA2375 SSA2410 SSA2430 SSA2440 SSA5070)
000015 ACTION(REP) RETAIN VALIDATE(BIND) ISOLATION(CS) ACQUIRE(USE) -
000016 RELEASE(COMMIT) EXPLAIN(NO)
000017 BIND PLAN(SSA3000) MEMBER(SSA3200) ACTION(REP) VALIDATE(BIND) -
000018 ISOLATION(CS) ACQUIRE(USE) RELEASE(COMMIT) EXPLAIN(NO) RETAIN
000019 BIND PLAN(SSA5000) MEMBER(SSA5000 SSA5070) ACTION(REP) ACQUIRE(USE) -
000020 VALIDATE(BIND) ISOLATION(CS) RETAIN RELEASE(COMMIT) EXPLAIN(NO)
000021 END
***** Bottom of Data *****
Aa A TCP/IP R 9 C 2 10:42 11/05/98

```

Review the following items and change them in the JCLs according to your installation's implementation of DB2.

- Verify that the STEPLIB DD refers to the correct DB2 load library.
 - Verify that the DBRMLIB DD statement refers to the SAMS:Compress DBRM dataset provided on the distribution tape.
 - Verify that the DB2 subsystem ID (DSN) is correct for the DB2 subsystem from which the ISPF compression analysis, Print Utility, or Immediate EDITPROC Implementation Utility are to be executed.
- 8 You can explicitly grant authorization to users of compression analysis. To do so you can either use SPUFI or the batch SQL processor to execute the following commands.

```

GRANT EXECUTE ON PLAN SSA0310 TO PUBLIC
GRANT EXECUTE ON PLAN SSA3000 TO PUBLIC
GRANT EXECUTE ON PLAN SSA5000 TO user ID

```

You do not have to "GRANT TO PUBLIC." An individual user is all right. User IDs are users who need to implement an immediate EDITPROC.

- 9 SAMS:Compress provides an Installation Verification Procedure (IVP) that ensures that the product was correctly installed and is able to compress and expand data using each of the standard routines. To run the IVP, simply use SPUFI to run the SQL statements contained in the SHRINK.DB2.JCL library member named IVP.

This procedure creates a table, inserts data into it, selects it, and drops it—using each of the SAMS:Compress standard compression routines. The process results in a return code of 100 or 0 from each of the SQL statements. If you do not get this result, contact SAMS:Compress Technical Support for assistance. If you get a -652 return code from any of the SQL statements, contact SAMS:Compress Technical Support immediately.

Upgrade Considerations

The new version of SAMS:Compress can be implemented at your installation regardless of which prior version of SAMS:Compress you have in operation. The new installation has no effect on implementing compression for any currently noncompressed DB2 table.

You are encouraged to replace any EDITPROCs from prior releases with those of the same name in this release, after appropriate testing. This gains you the benefits of any maintenance included in this release, for example, to S30INTEL. You can use the “Recreate Unique EDITPROC Panel” to update any Optimal routines for this release.

SAMS:Compress compression routines currently specified for any table continue to operate in the same manner. And the SAMS:Compress Technical Support Center continues to offer the same support for these routines. To see the advantages that the new compression routines offer over prior versions, you must specify the new EDITPROC compression routine on any table currently compressed. In order to implement an EDITPROC routine, you need to perform the same steps you used when you specified the initial compression routine - you must unload, drop, and recreate the table.

Third Party Program Installation Considerations

To use SAMS:Compress compression routines with third party programs outside of DB2, see “Third Party Program Capability” on page 65 for the optional installation schemes.

Messages

This appendix contains all the error and Informational messages generated by SAMS:Compress. The types of messages generated are:

- Batch Program Error Messages
- Online Error Messages
- Online Informational Messages

Each message is identified by a message number consisting of two parts. The first is an alpha character identifier; DCA and SSB (batch program error), SSE (online error), and SSI (informational). The second contains an alphanumeric identifier. All messages are listed by message number.

Batch Programs Error Messages

This section contains the SAMS:Compress error messages that can occur during batch jobs processing. In some messages, the message issued by SAMS:Compress are preceded by an eight-character module name (GDACxxx, for example) and the message number is preceded by an 8-character module name (DCAAnnnn).

DCA0001 **COMMAND ERROR - EXECUTION SUPPRESSED**

Explanation: An error was detected by the command processor while it was attempting to process the input commands. The exact nature of the error is detailed by a previous error message.

Action: Resubmit the job with the correct command specifications.

DCA0002 UNABLE TO OPEN INPUT FILE

Explanation: Unable to open the SYSIN dd statement that should contain command statements.

Action: See other associated messages for reasons why the open failed.

DCA0003 CHECK FOR MISSING SYSIN DD STATEMENT

Explanation: Could not find the SYSIN dd statement that should contain command statements.

Action: Verify that command statements do exist. If this function is being invoked from a multi-step JCL procedure, ensure that the commands are being input to the correct step.

This may require a dd statement to direct it to the proper step (for example:

```
//DCA.SYSIN DD *)
```

DCA0004 COMMAND REJECTED:COMMAND NAME CONTAINS AN INVALID ALPHA CHARACTER (xxx)

Explanation: The utility has rejected a command entered in the input stream (DDNAME = SYSIN) because of a syntax error.

The command name (xxx) is not a valid command name for the function being executed.

Action: Verify that the following conditions are correct:

- A valid command name is entered on the command.
- The command name is spelled correctly.
- The function associated with the command is being invoked properly.
- No characters appear before the command name in the input stream record.

Processing continues with the next input command. Correct the command in error and rerun.

DCA0005 **COMMAND REJECTED:CHARACTER (x) AT COLUMN (nn) IS AN INVALID CHARACTER FOR SAMS:Compress COMMANDS**

Explanation: Command entered in the input stream (DDNAME = SYSIN) was rejected because of a syntax error. The character (x) is not valid for command data.

Action: Remove the character (x) from the command or replace it with a valid character.

Processing continues with the next input command. Correct the command in error and rerun.

DCA0006 **COMMAND REJECTED:TRAILING COMMA ON PREVIOUS COMMAND INDICATES CONTINUATION REQUIRED, BUT NO CONTINUATION LINE WAS FOUND**

Explanation: Command entered in the input stream (DDNAME = SYSIN) was rejected because of a syntax error.

Action: Verify that continuation rules for the commands entered have been followed. Processing continues with the next input command. Correct the command in error and rerun.

DCA0007 **COMMAND REJECTED:SAMS:Compress DOES NOT ALLOW NON-BLANK CHARACTERS IN COLUMN 72 OF COMMAND STATEMENTS**

Explanation: Command entered in the input stream (DDNAME = SYSIN) was rejected because of syntax error. Continuation rules do not require a non-blank character in column 72. They also do not allow the column to contain data. Columns 73 through 80 of the commands entered are not used.

Action: Verify that continuation rules for the commands entered have been followed. Processing continues with the next input command. Correct the command in error and rerun.

DCA0008 **COMMAND REJECTED:COMMAND NAME (nnn)
IS LONGER THAN MAXIMUM NAME LENGTH OF
(xxx)**

Explanation: Command entered in the input stream (DDNAME =
SYSIN) was rejected because of a syntax error.

Action: Verify that the following conditions are correct.

- Command names must be less than (xxx) characters.
- A valid command name is entered on the command.
- The command name is spelled correctly.
- No characters appear before the command name in the input stream record.

Processing continues with the next input command. Correct the command in error and rerun.

DCA0009 **COMMAND REJECTED:ERROR IN PARAMETER
AT COLUMN (nn):
NULL PARAMETERS ARE NOT ALLOWED,
REPLACE WITH PROPER VALUE**

Explanation: Command entered in the input stream (DDNAME =
SYSIN) was rejected because of a syntax error.

A command parameter that requires a value to be supplied has not been given a valid value.

Action: If the data appears to be correct on the command, verify that the following conditions are correct.

- The parameter name is spelled correctly.
- The continuation rules have been followed if the command is continued on multiple lines.
- No spaces are found within the command parameter data unless they are within quotation marks.
- The command name is the one you intended to execute.

Processing continues with the next input command. Correct the command in error and rerun.

DCA0010 **PARAMETER DATA STARTING IN COLUMN (nn)
EXCEEDS MAXIMUM LENGTH FOR THIS
PARAMETER (xx)**

Explanation: Command entered in the input stream (DDNAME = SYSIN) was rejected because of a syntax error. (xx) indicates that the column of the first character or the column just prior to the parameter is in error. If more than one command statement was used for the command, the error pertains to the command statement printed just prior to this error message. The command entered contains an invalid parameter name.

Action: Verify that the following items are correct.

- The parameter name is spelled correctly.
- The continuation rules have been followed, if the command is continued on multiple lines.
- No spaces are found within the command parameter data unless they are within quotation marks.
- The command name is the one you intended to execute.

Processing continues with the next input command. Correct the command in error and rerun.

DCA0011 **COMMAND REJECTED: SUPPLIED PARAMETER
(nnn) IS NOT A VALID PARAMETER FOR THE
COMMAND NAME SPECIFIED**

Explanation: Command entered in the input stream (DDNAME = SYSIN) was rejected because of a syntax error.

Action: The command entered contains an invalid parameter name. Verify that the following items are correct.

- The parameter name is spelled correctly.
- The continuation rules have been followed, if the command is continued on multiple lines.
- No spaces are found within the command parameter data unless they are within quotation marks.
- The command name is the one you intended to execute.

Processing continues with the next input command. Correct the command in error and rerun.

DCA0013 **COMMAND REJECTED: PARAMETER (nnn) HAS BEEN SPECIFIED MORE THAN ONCE**

Explanation: Command entered in the input stream (DDNAME = SYSIN) was rejected because of a syntax error. The command entered contains the same parameter specified twice.

Action: Verify the following.

- The parameter name is spelled correctly.
- The continuation rules have been followed, if the command is continued on multiple lines.
- No spaces are found within the command parameter data unless they are within quotation marks.

Processing continues with the next input command. Correct the command in error and rerun.

DCA0014 **COMMAND REJECTED: KEYWORD PARAMETER nnn HAS BEEN SPECIFIED WITHOUT A VALUE**

Explanation: Command entered in the input stream (DDNAME = SYSIN) was rejected because of a syntax error. A command parameter that requires a value to be supplied has not been given a valid value.

Action: If the data appears to be correct on the command, verify that the following conditions are correct.

- The parameter name is spelled correctly.
- The continuation rules have been followed, if the command is continued on multiple lines.
- No spaces are found within the command parameter data unless they are within quotation marks.
- The command name is the one you intended to execute.

Processing continues with the next input command. Correct the command in error and rerun.

DCA0017 **COMMAND REJECTED: KEYWORD PARAMETER
nnn CANNOT SPECIFY A LIST OF VALUES:
SPECIFY ONLY ONE VALUE AND REMOVE
PARENTHESES**

Explanation: Command entered in the input stream (DDNAME =
SYSIN) was rejected because of a syntax error.

The utility allows only one specification of value data for this parameter. The parentheses imply that more than one value is supplied.

Action: Remove the parentheses and specify only one value.

If the data appears to be correct on the command, verify that the following conditions are correct.

- The parameter name is spelled correctly.
- The continuation rules have been followed, if the command is continued on multiple lines.
- No spaces are found within the command parameter data unless they are within quotation marks.
- The command name is the one you intended to execute.

Processing continues with the next input command. Correct the command in error and rerun.

DCA0018 **COMMAND REJECTED: KEYWORD PARAMETER
nnn REQUIRES A NUMERIC VALUE; VALUE
SPECIFIED IS NOT NUMERIC**

Explanation: Command entered in the input stream (DDNAME =
SYSIN) was rejected because of a syntax error.

Command parameter (nnn) requires numeric data as values specified for it. One or more values are not numeric.

Action: If the data appears to be correct on the command, verify that the following conditions are correct.

- The parameter name is spelled correctly.
- The continuation rules have been followed, if the command is continued on multiple lines.
- The command name is the one you intended to

execute.

Processing continues with the next input command. Correct the command in error and rerun.

DCA0019 **COMMAND REJECTED: SIMPLE PARAMETER (nnn) IS FOLLOWED BY OTHER THAN A COMMA OR A SPACE; NO VALUES CAN BE ASSOCIATED WITH THIS SIMPLE PARAMETER**

Explanation: Command entered in the input stream (DDNAME = SYSIN) was rejected because of a syntax error.

A SIMPLE parameter has no values. The parameter (nnn) is a SIMPLE parameter but the user has tried to enter a value for it. The parameter name must be followed by either a comma or a space.

Action: If the data appears to be correct on the command, verify that the following conditions are correct.

- The parameter name is spelled correctly.
- The continuation rules have been followed, if the command is continued on multiple lines.

Processing continues with the next input command. Correct the command in error and rerun.

DCA0020 **COMMAND REJECTED: PARAMETER (nnn) HAS A NUMERIC FIELD VALUE GREATER THAN (vvv)**

Explanation: Command entered in the input stream (DDNAME = SYSIN) was rejected because of a syntax error.

The parameter (nnn) accepts a numeric value as input. The value cannot be greater than (vvv). This value was specified incorrectly.

Action: If the value (vvv) appears to be correct on the command, verify that the following conditions are correct.

- The parameter name is spelled correctly.
- The continuation rules have been followed, if the command is continued on multiple lines.
- The command name is the one you intended to execute.

Processing continues with the next input command. Correct the command in error and rerun.

DCA0030 **COMMAND REJECTED: (kkk) IS NOT A VALID COMMAND NAME**

Explanation: Command entered in the input stream (DDNAME = SYSIN) was rejected because of a syntax error.

The command name (kkk) is not a valid command name for the function being executed.

Action: Verify that the following conditions are correct.

- A valid command name is entered on the command.
- The command name is spelled correctly.
- The function associated with the command is being invoked properly.
- No characters appear before the command name in the input stream record.

Action: Processing continues with the next input command.
Correct the command in error and rerun.

DCA0031 **NO VALID COMMANDS WERE ENTERED**

Explanation: The command processor has found no valid commands to return to the processing routines. Either all commands were in error, or no commands were supplied through the SYSIN dd statement.

Action: Supply at least one valid command and rerun.

DCA0033 **COMMAND REJECTED: PARAMETER (vvv) IS NOT SPECIFIED AND IS REQUIRED FOR COMMAND (kkk)**

Explanation: Command entered in the input stream (DDNAME = SYSIN) was rejected because of a syntax error.

The command name (kkk) has an associated required parameter (vvv) and it is not specified.

Action: Supply the (vvv) parameter if it is missing. If it appears to be on the command, verify that the following conditions are correct.

- The parameter name is spelled correctly.
- The continuation rules have been followed, if the

command is continued on multiple lines.

- No spaces are found within the command parameter data unless they are within quotation marks.
- The command name is the one you intended to execute.

Processing continues with the next input command. Correct the command in error and rerun.

SSA2000A **MAXIMUM RECORD SIZE EXCEEDS MAXIMUM ALLOWED WITH COMPRESSION**

MAXIMUM RECORD SIZE: nnnnn

MAXIMUM ALLOWED WITH COMPRESSION: nnnnn

Explanation: The table being analyzed is defined with one of the following.

- LONG VARCHAR
- LONG VARGRAPHIC
- A mixture of fixed and varying length columns

These table definitions cause the table to exceed the maximum record length if compression were implemented. The reason this happens is that an EDITPROC requires 10 bytes of a row, thus, there are 10 bytes less for data within the row.

Action: Change the data or table definition to a maximum value that is less than that shown for MAXIMUM ALLOWED WITH COMPRESSION. Refer to the the IBM manual DB2 SQL Reference for more information.

SSA50001I **(kkk) - COMMAND ACCEPTED.**

Explanation: Informational message confirming that the command shown was accepted by the utility.

Action: None.

SSA50002I (kkk) - COMMAND COMPLETE, RETURN CODE = (xx)

Explanation: Informational message showing that the utility has completed the command and generated the return code shown.

Action: None.

SSA50003I COMMAND FLUSHED AND NOT EXECUTED.

Explanation: The current command was not executed due to an error in a previous command.

Action: Resubmit the job with the correct command specifications.

SSA50004E CREATOR / TABLE NAME INVALID OR NOT FOUND: (xxxxxxxx)

Explanation: The utility was not able to either locate the table shown or it could not validate the creator that you specified.

Action: Resubmit the job with the correct creator or table name specified.

SSA50005I EXECUTION COMPLETE, RETURN CODE = (xx)

Explanation: Informational message confirming that the utility has completed all processing. The highest return code encountered is xx.

Action: None.

SSA50008E COMMAND (OR PARAMETERS) INVALID.

Explanation: Command rejected because command parameters specified are invalid or incorrect.

Action: Correct the command parameters specified and resubmit the job.

SSA50009I COMMANDS FOLLOWING WILL BE VALIDATED AND SIMULATED, BUT THE REQUESTED ACTION WILL NOT BE TAKEN.

Explanation: Informational message confirming that the commands shown will be run in simulate mode. No action taken.

Action: None.

SSA50012T AN SQL ERROR HAS OCCURRED.

Explanation: An SQL error was encountered while processing your command.

Action: Refer to the IBM manual DB2 SQL Message and Codes for more information. Contact SAMS:Compress support if you are unable to resolve the error.

SSA50016S DB2 SUBSYSTEM (xxxx) NOT FOUND OR UNAVAILABLE.

Explanation: During the execution of your job, the utility was unable to locate the DB2 subsystem specified.

Action: Ensure that the DB2 subsystem is active and that you have spelled it correctly. If unable to resolve the error, contact SAMS:Compress support.

SSA50020S NON-RECOVERABLE ERROR IN MODULE (modname).

Explanation: A nonrecoverable error occurred while the utility was processing your request.

Action: See the accompanying message(s).

SSA50021S PLEASE CALL SUPPORT FOR ASSISTANCE.

Explanation: The utility experienced a previously identified error.

Action: Contact the SAMS:Compress Technical Support Center.

SSA50031I SIMULATE - COMMAND SIMULATION IS (on/off)

Explanation: Informational message informing you that simulation mode is either active or inactive. When simulation is active, the utility will not implement any of the commands.

Action: None.

SSA50071E INVALID COLUMN TYPE, TYPE=

Explanation: The column type you attempted to add to your table is not a valid DB2 specification.

Action: Verify the table type specification and resubmit the job.

SSA50072E THE COLUMN CAN NOT BE ADDED BECAUSE THE DB2 PAGE SIZE WOULD BE EXCEEDED.

Explanation: DB2 requires that an entire row be contained on a single DB2 page. The column that you are attempting to add would cause the combined lengths of the new rows to exceed the DB2 limit.

Action: Change the type of column or its size to reduce the length of the new column and resubmit the job. Or you can recreate the table with a larger DB2 page size. This will allow you to increase the length of each row with the addition of the new column.

SSA50073I COLUMN (xxxxxxx) HAS A SIZE = (xxx) BYTES, AND IS OF TYPE = (xxx)

Explanation: Informational message that displays the characteristics of the new column that you are adding.

Action: None.

SSA50075E FOR= IS INVALID FOR A NON-CHARACTER COLUMN.

Explanation: You may not specify FOR= with the type of column specified in TYPE=. The FOR= parameter may be used only with a character type column.

Action: Omit the parameter entirely or specify a character type column and resubmit the job.

SSA50076E INVALID VALUE FOR PARAMETER: (xxxxxxx)

Explanation: The value that you specified for the parameter (xxxxxxx) is invalid.

Action: Correct the specification and resubmit the job.

SSA50077E FIELDPROC MAY NOT BE SPECIFIED WITH NOTNULL=Y

Explanation: DB2 does not allow you to specify a field procedure name when NOTNULL is set to Y.

Action: Omit the field procedure from the column or change the setting of NOTNULL and resubmit the job.

SSA50079E COLUMN LENGTH SPECIFICATION INCORRECT.

Explanation: The value specified for the LENGTH= parameter in your job is incorrect for the type of column you are adding.

Action: Correct or eliminate the value and resubmit the job. (Refer to [“Using Additional Programs on Compressed Tables”](#) on page 59 for help in determining the correct column length.)

SSA50090I DB2 CATALOG INFORMATION FOR TABLE:

Explanation: Informational message identifying the table specifications when the utility began processing the command.

Action: None.

SSA50091I GENERATED SQL STATEMENT FOLLOWS, OF LENGTH= (xxx)

Explanation: Informational message indicating the length of the SQL statement generated during processing. (The SQL statement generated will follow this message.)

Action: None.

SSA50099E AN UNEXPECTED ERROR HAS OCCURRED.

Explanation: During utility processing, an unexpected error has occurred.

Action: Contact the SAMS:Compress Technical Support Center.

SSA50700I (diagnostic message)

Explanation: This message identifies the start of a group of support and diagnostic messages.

Action: Refer to the subsequent messages displayed.

SSA50701E CREATOR / TABLE NAME INCORRECT OR INVALID.

Explanation: The creator or table name you specified was either incorrect or invalid.

Action: Correct these parameters and resubmit the job.

SSA50702E EDITPROC IMPLEMENTATION PROCESS ONLY SUPPORTS TABLES. IT DOES NOT SUPPORT ALIASES OR VIEWS.

Explanation: The Immediate EDITPROC Implementation Utility cannot be used on alias or view definitions.

Action: Specify a table name and resubmit the job.

SSA50704E ADD REJECTED, TABLE ALREADY HAS EDITPROC (xxxxxxx)

Explanation: The utility was unable to add a new EDITPROC to your table because the table currently uses the (xxxxxxx) EDITPROC.

Action: Use the CHANGEEDIT command to implement the new EDITPROC on your table.

SSA50705E ADD REJECTED, RECORD LENGTH WITH EDITPROC WOULD EXCEED MAXIMUM ALLOWED BY DB2.

Explanation: The utility was unable to implement the EDITPROC you specified because an EDITPROC requires that the row have at least 10 bytes remaining within the DB2 page size.

Action: Recreate the table with a larger DB2 page size or reduce the size or number of columns so that at least 10 bytes are available for the EDITPROC.

SSA50706E FUNCTION REJECTED, TABLE DOES NOT HAVE AN EDITPROC.

Explanation: The utility was unable to remove the EDITPROC from the table because the DB2 table you specified does not contain an EDITPROC.

Action: Verify that the table name specified is correct and resubmit the job.

SSA50707E REMOVE OF (xxxxxxx) REJECTED, TABLE HAS EDITPROC (yyyyyyy).

Explanation: The utility was unable to remove the EDITPROC (xxxxxxx) from your table.

Action: If you specified an EDITPROC for the REMOVEEDIT command, it is incorrect. If you did not, an error has occurred within the utility and you should contact SAMS:Compress support.

SSA50708E THE REQUESTED EDITPROC IS NOT SUPPORTED FOR THIS FUNCTION: (xxxxxxx).

Explanation: The utility was unable to add the EDITPROC (xxxxxxx) specified because it is not a valid compression routine.

Action: Verify the name of the EDITPROC and resubmit the job or specify the NOEDITCHECK parameter to disable the EDITPROC validation.

SSA50709I END OF DATA ON IMPLEMENTATION WORK TABLE.

Explanation: Informational message following a group of support and diagnostic messages.

Action: None.

SSA50710I A STOP OR START DATABASE / TABLESPACE RESULTED IN THE FOLLOWING MESSAGE(S):

Explanation: A start or stop command ended in an error. This message identifies the start of a group of support and diagnostic messages.

Action: Refer to the subsequent messages displayed.

SSA50711E VALIDPROC (S32VALID) ALREADY SET.

Explanation: The last attempt to run the utility resulted in a failure.

Action: Refer to subsequent messages.

SSA50712I IMPLEMENTATION WORK TABLE ALREADY EXISTS.

Explanation: Informational message informing you that a previous copy of a work table exists.

Action: See accompanying message.

SSA50713T A PRIOR IMPLEMENTATION FAILURE HAS NOT BEEN RESOLVED.

Explanation: The last attempt to run the utility resulted in a failure that was not corrected by the utility.

Action: Specify the option RESETFAILURE in your utility and resubmit the job, or set the Reset Prior Failure field to Y on the ISPF panel. If you continue to receive this message, contact SAMS:Compress support. Have the print from the failed utility execution available, as well as the output from the DB2 Utility Repair Diagnose.

SSA50714E THE REQUESTED FUNCTION REQUIRES THE WILLRELOAD OPTION INDICATING YOU WILL RELOAD YOUR DATA.

Explanation: The utility detected that the function being performed, considering the EDITPROC involved, could render the table's data as unusable unless you reload and recompress the data from an uncompressed backup.

Action: Either take the steps necessary to preserve your data as listed under the WILLRELOAD option or do not perform this function.

SSA50715E DBD DIRECTORY UPDATES AND CATALOG BACK OUT HAVE BOTH FAILED.

Explanation: An error occurred while the utility was synchronizing the DB2 directory with the DB2 catalog. The utility then attempted to back out the DB2 catalog updates when a second error occurred.

Action: Contact SAMS:Compress Support. Have the print from the failed utility execution available, as well as the output from the DB2 Utility Repair Diagnose.

SSA50716E UNABLE TO CREATE THE IMPLEMENTATION WORK TABLE.

Explanation: The utility was unable to create the required implementation work table.

Action: Review the accompanying SQL error code and messages to decide if the WORKTS= parameter should be specified. Contact SAMS:Compress Technical Support if you are unable to resolve this error.

SSB1001 RCREATOR MUST BE SPECIFIED WITH LEVEL 3.

Explanation: The utility failed to respond because it requires that a compression analysis repository be specified in the ANALYZE command.

Action: Supply the (xxxx) parameter to produce the report.

SSB1002 RPREFIX MUST BE SPECIFIED WITH LEVEL 3

Explanation: The utility failed to respond because it requires that a compression analysis repository be specified in the ANALYZE command.

Action: Supply the (xxxx) parameter to produce the report.

SSB1003 OPTIMIZE NOT ALLOWED WITH LEVEL 1, IGNORED

Explanation: The utility failed to respond because, when producing the Basic Functions report (specified by LEVEL=1), it could not produce the optimal compression routine. The Basic Functions report was created with no optimal routines.

Action: Either change LEVEL to 3 to produce the Advanced Functions report (and specify RCREATOR and RPREFIX), or remove OPTIMIZE from the ANALYZE command.

SSB1004 CREATOR (xxxx) AND PREFIX (xxxx) REPOSITORY TABLES NOT FOUND IN CURRENT DB2 SYSTEM

Explanation: The utility failed to respond because the existing compression analysis repository must be specified in the ANALYZE command.

Action: Reenter the creator and prefix to correspond to an existing repository.

SSB1005 TIME PARAMETER VALUE EXCEEDED FOR CURRENT TABLE

Explanation: The utility compression analysis processing for the current table exceeded the value specified for the TIME= parameter. Results of the analysis are based on the number of rows analyzed in the given time frame, not on the number specified in the parameter.

Action: No action is required, however, increasing the time value allows the utility to analyze larger number of rows which will produce more accurate results.

Online Error Messages

This section contains the SAMS:Compress online error messages.

SSE001 (xxxx) is REQUIRED

Explanation: The utility has failed to respond because a value is required for the (xxxx) field.

Action: Specify a value for the (xxxx) field.

SSE002 (xxxx) is INVALID

Explanation: The utility has failed to respond because the value entered in the (xxxx) field is not valid.

Action: Reenter the value contained within the (xxxx) field.

SSE003 (xxxx) MUST BE NUMERIC, CONSISTING OF CHARS (0..9)

Explanation: The utility has failed to respond because the value entered in the (xxxx) field is not composed exclusively of numeric characters.

Action: Reenter the value contained within the (xxxx) field using only numeric characters (0 through 9).

SSE004 (xxxx) MUST BE ALPHABETIC, CONSISTING OF CHARS (A..Z)

Explanation: The utility has failed to respond because the value entered in the (xxxx) field is not composed exclusively of alpha characters.

Action: Reenter the value contained within the (xxxx) field using only alpha characters ("a" through "z").

SSE200A S30INTEL COMPRESSION ANALYSIS RETURN CODE: (xxxxxxxx)

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

SSE200B S30INTEL COMPRESSION ANALYSIS RETURN CODE: (xxxxxxxx)

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

SSE200C S30INTEL COMPRESSION ANALYSIS RETURN CODE: (xxxxxxxx)

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

SSE200D S30INTEL COMPRESSION ANALYSIS RETURN CODE: (xxxxxxxx)

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

SSE200E S30INTEL COMPRESSION ANALYSIS RETURN CODE: (xxxxxxxx)

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

SSE200F S30INTEL COMPRESSION ANALYSIS RETURN CODE: (xxxxxxxx)

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

SSE200N **INTERNAL ERROR - ROUTINE TABLE SPACE EXCEEDED - OPTIMAL**

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

SSE200O **INTERNAL ERROR - ROUTINE TABLE SPACE EXCEEDED - EDITPROC**

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

SSE200P **INTERNAL ERROR - COMPILE CODE BUILD RETURN CODE: (xxxxxxx)**

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

SSE202A **COMPRESSION ANALYSIS RETURN CODE: (xxxxxxx)**

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

SSE202B **COMPRESSION ANALYSIS RETURN CODE: (xxxxxxx)**

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

**SSE202C COMPRESSION ANALYSIS RETURN CODE:
(xxxxxxxx)**

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

**SSE202D COMPRESSION ANALYSIS INPUT/OUTPUT
LENGTH MISMATCH**

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

**SSE202E COMPRESSION ANALYSIS DATA COMPARE
MISMATCH**

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

**SSE202F OUTPUT AREA EXCEEDED ON COMPRESS,
ANALYSIS TERMINATED**

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

**SSE202G OUTPUT AREA EXCEEDED ON DECOMPRESS,
ANALYSIS TERMINATED**

Explanation: The utility compression analysis facility has detected an anomaly within your DB2 table. This code indicates that an error may reside within the DB2 table under analysis.

Action: Note the eight-character error code and contact the SAMS:Compress Technical Support Center.

SSE202X CALL SAMS:Compress SOFTWARE SUPPORT

Explanation: An unexpected error has been detected.

Action: Contact the SAMS:Compress Technical Support Center.

SSE202Z INTERNAL ERROR DETECTED, PROCESSING TERMINATED

Explanation: An unexpected error has been detected.

Action: Contact the SAMS:Compress Technical Support Center.

SSE210A USER LEVEL (xxxx) 1=BEGINNER, 2=INTERMEDIATE, 3=ADVANCED

Explanation: The utility has failed to respond because an inappropriate user level was specified within the CLIST member used to activate the compression analysis facility.

Action: Locate the CLIST member which failed to function properly (SDB2—Version 3.2, SDB2BASIC - Basic Functions, or SDB2ADV- Advanced Functions) from within the TSO CLIST library. Change the following line within the Update/ Define Global Variables section by setting the value of x equal to 1 within SDB2BASIC or equal to 3 within SDB2ADV or SDB2.

```
SET &FZUSRLVL = &STR(x)
```

SSE210B DB2 SUBSYSTEM IS A REQUIRED FIELD

Explanation: The utility failed to respond because the SUBSYS field does not contain a value on the Initial Panel.

Action: Enter a valid subsystem value in the SUBSYS field on the Initial Panel.

SSE210C DB2 SUBSYSTEM (xxxx) MUST BE A VALID LABEL

Explanation: The utility failed to respond because the subsystem (xxxx) is not a valid label.

Action: Reenter the subsystem value in the SUBSYS field on the Initial Panel to conform to DB2 naming conventions.

SSE212A PLEASE RESPOND "Y" FOR YES, "N" FOR NO

Explanation: The utility failed to respond because a value other than Y (YES) or N (NO) was entered into the first field on the Repository Setup Panel.

Action: Reenter your response on the first field using either Y or N.

SSE212B CREATOR IS A REQUIRED FIELD

Explanation: The utility failed to respond because the Creator field on the Initial Panel does not contain a value.

Action: Enter a valid creator value in the Creator field on the Initial Panel.

SSE212C CREATOR (xxxx) MUST BE A VALID LABEL

Explanation: The utility failed to respond because the value entered into the Creator field on the Initial Panel is not valid.

Action: Reenter the creator ID in the Creator field on the Initial Panel to conform to DB2 naming conventions.

SSE212D PREFIX (xxxx) IS A REQUIRED FIELD

Explanation: The utility failed to respond because the Prefix field on the Initial Panel does not contain a value.

Action: Enter a valid prefix name in the Prefix field on the Initial Panel.

SSE212E PREFIX (xxxx) MUST BE A VALID LABEL

Explanation: The utility failed to respond because the value entered into the Prefix field on the Initial Panel is not valid.

Action: Reenter the prefix name in the Prefix field on the Initial Panel to conform to DB2 naming conventions.

SSE212F (xxxx) TABLES ALREADY EXIST, PRESS END TO USE THEM

Explanation: The utility failed to respond because a repository already exists that uses the creator ID and prefix name entered on the Repository Creation Panel.

Action: If you wish to use the repository defined by the current creator ID and prefix name, press the <END> key to return to the Initial Panel and enter these same values into the Creator and Prefix fields. If you wish to define a new repository, reenter either the creator ID or the prefix name to different values.

SSE212G (xxxx) TABLES SHOULD BE CREATED

Explanation: The utility failed to respond because the repository tables identified by the creator ID and prefix name on the Repository Setup Panel do not currently exist.

Action: This message is displayed to let you confirm the creation of the new repository tables. Press the <ENTER> key and the table creation will begin. If you change your mind and do not wish to create the indicated repository, either change the creator ID or prefix name or press the <END> key to return to the Initial Panel.

SSE212H (xxxx) TABLES WERE NOT CREATED

Explanation: The utility failed to respond because some problem was encountered during the table creation process.

Action: Review the accompanying DB2 error messages.

SSE216A STOGROUP IS A REQUIRED FIELD

Explanation: The utility failed to respond because the value in the STOGROUP NAME field was erased on the Repository Specification Panel.

Action: Enter a storage group name in the STOGROUP NAME field.

SSE216B STOGROUP (xxxx) MUST BE A VALID LABEL

Explanation: The utility failed to respond because the storage group name entered in the STOGROUP NAME field on the Repository Specification Panel is not valid.

Action: Reenter the storage group name to conform to DB2 naming conventions.

SSE216C STOGROUP (xxxx) ALREADY EXISTS

Explanation: The utility failed to respond because the storage group name specified in the STOGROUP NAME field on the Repository Specification Panel already exists.

Action: If you wish to create the repository with the currently specified storage group, press the <END> key to return to the Repository Creation Panel. Then enter a D on the line containing the storage group name. The next panel displayed will allow you to maintain the same storage group name and change the database and tablespace names.

SSE216D DATABASE IS A REQUIRED FIELD

Explanation: The utility failed to respond because the value in the DATABASE NAME field was blank on the Repository Specification Panel.

Action: Enter a database name in the DATABASE NAME field.

SSE216E DATABASE (xxxx) MUST BE A VALID LABEL

Explanation: The utility failed to respond because the database name entered in the DATABASE NAME field on the Repository Specification Panel is not valid.

Action: Reenter the database name in the DATABASE NAME field on the Repository Specification Panel to conform to DB2 naming conventions.

SSE216F DATABASE (xxxx) ALREADY EXISTS

Explanation: The utility failed to respond because the database specified in the DATABASE NAME field on the Repository Specification Panel already exists.

Action: If you wish to create the repository with the currently specified database, press the <END> key to return to the Repository Creation Panel. Then enter a T on the line containing the database name. The next panel displayed will allow you to maintain the same storage group and database and change the tablespace name.

SSE216G TABLESPACE IS A REQUIRED FIELD

Explanation: The utility failed to respond because the value in the TABLESPACE NAME field was blank on the Repository Specification Panel.

Action: Enter a tablespace name in the TABLESPACE NAME field.

SSE216H TABLESPACE (xxxx) MUST BE A VALID LABEL

Explanation: The utility failed to respond because the entry in the TABLESPACE NAME field on the Repository Specification Panel is not valid.

Action: Reenter the tablespace name to conform to DB2 naming conventions.

SSE216I TABLESPACE (xxxx) ALREADY EXISTS

Explanation: The utility failed to respond because the tablespace specified in the TABLESPACE NAME field on the Repository Specification Panel already exists.

Action: To create the repository with the currently specified tablespace, press the <END> key to return to the Repository Creation Panel. Then enter a C on the line containing the tablespace name. The repository will be created using the storage group, database, and tablespace specified.

SSE216J VOLUMES IS A REQUIRED FIELD

Explanation: The utility failed to respond because no value was specified in the STOGROUP VOLUMES field on the Repository Specification Panel.

Action: Enter volume names in the STOGROUP VOLUMES field on the Repository Specification Panel.

SSE216K PRIQTY IS A REQUIRED FIELD

Explanation: The utility failed to respond because no value was specified in the TABLESPACE PRIQTY field on the Repository Specification Panel.

Action: Enter a primary value on the Repository Specification Panel.

SSE216L PRIQTY MUST BE NUMERIC

Explanation: The utility failed to respond because the priority value enter in the TABLESPACE PRIQTY on the Repository Specification Panel was not exclusively numeric characters.

Action: Reenter the primary quantity using numeric characters 0 through 9.

SSE216M SECQTY MUST BE NUMERIC

Explanation: The utility failed to respond because the security value enter in the TABLESPACE SECQTY on the Repository Specification Panel was not exclusively numeric characters.

Action: Reenter the secondary quantity using numeric characters 0 through 9.

SSE216N VCAT IS A REQUIRED FIELD

Explanation: The utility failed to respond because no value was specified in the STOGROUP VCAT field on the Repository Specification Panel.

Action: Enter a valid VCAT value.

SSE310A NO TABLES FOUND THAT MATCH YOUR SELECTION CRITERIA

Explanation: The utility failed to respond because the patterns for the database, creator ID, and table name specified on the Database\Creator\Table Specification Panel do not match any tables within the current subsystem.

Action: Reenter some or all of the patterns to broaden the scope.

SSE310X SAMS:Compress PRODUCT EVALUATION PERIOD HAS EXPIRED

Explanation: The utility failed to respond because the evaluation period has expired.

Action: Contact your marketing representative to request an extension of the evaluation period.

SSE314A THIS COMMAND PREFIX IS ONLY VALID AFTER ANALYSIS

Explanation: The utility failed to respond because an A, B (Advanced Functions), or a D (Basic Functions) can be specified only on the Table Selection List and Compression Statistics for a table that has been analyzed.

Action: Enter an S to conduct compression analysis of the table.

SSE314B ERROR DURING ANALYSIS, RESULTS UNAVAILABLE

Explanation: An error has occurred which prevented successful analysis.

Action: Rerun the analysis as a batch job to obtain diagnostic information and then contact SAMS:Compress Technical Support.

SSE314N NO RESULTS AVAILABLE FOR DISPLAY

Explanation: No results are available for display from either the batch or online analysis function.

Action: Wait for the batch analysis to complete processing, or perform the online analysis of the table.

SSE350A ALL COLUMNS APPLY TO THIS EDITPROC

Explanation: This is an Informational message that indicates that this EDITPROC will process all columns in the table. Only certain optimal EDITPROCs do not process all columns.

Action: None.

SSE350B CREATE EDITPROC FOR THIS ROUTINE IS NOT VALID

Explanation: The utility failed to respond because the C function is used only to create unique EDITPROCs for optimal routines.

Action: A unique EDITPROC is not necessary for the standard routines. Select an optimal routine to create a unique EDITPROC or use a standard compression routine for implementation on the table.

SSE350C UNIQUE EDITPROC DOES NOT CURRENTLY EXIST FOR THIS ROUTINE

Explanation: The utility failed to respond because no unique EDITPROCs have been created using this optimal routine on the above table.

Action: If you are trying to locate a unique EDITPROC for this table, select function C. This will provide a complete list of all unique EDITPROCs created in this repository. If the EDITPROC is not on the list, search the other repositories.

SSE350E IMMEDIATE EDITPROC IMPLEMENTATION ENDED WITH AN ERROR

Explanation: An error occurred while attempting to immediately implement an EDITPROC.

Action: See accompanying message(s).

SSE350F IMMEDIATE EDITPROC IMPLEMENTATION HAS FAILED

Explanation: The utility failed to implement the EDITPROC on your table due to a utility failure.

Action: See accompanying message(s).

SSE350M IMMEDIATE EDITPROC IMPLEMENTATION ONLY SUPPORTS FLEXIBLE ROUTINES

Explanation: The ISPF panels allow you to immediately implement only flexible routines.

Action: Specify a flexible routine to implement.

SSE350N IMMEDIATE EDITPROC IMPLEMENTATION TERMINATED - NOT CONFIRMED

Explanation: On the Confirm Immediate EDITPROC Implementation Panel, you replied N to the confirmation or you pressed the <END> key.

Action: None. If you do wish to immediately implement the EDITPROC, reissue your request and reply Y to confirm.

SSE350S IMMEDIATE EDITPROC IMPLEMENTATION COMPLETED SUCCESSFULLY

Explanation: Informational message informing you that the EDITPROC has been implemented successfully.

Action: None.

SSE702 THE EDITPROC NAME IS REQUIRED IN ORDER TO CREATE AN EDITPROC

Explanation: The utility failed to respond because a routine name was not entered in the first field on the Create Unique EDITPROC Panel.

Action: Enter a routine name for the unique EDITPROC in the first field.

SSE703 THE EDITPROC NAME MUST CONFORM TO MEMBER NAMING CONVENTIONS

Explanation: The utility failed to respond because the routine name specified in the first field on the Create Unique EDITPROC Panel does not conform to member naming conventions.

Action: Record the accompanying MVS message and reenter the routine name addressing the MVS error message.

SSE704 EDITPROC NAME MUST NOT ALREADY EXIST IN THE SAMS:Compress REPOSITORY

Explanation: The utility failed to respond because the routine name specified in the first field on the Create Unique EDITPROC Panel already exists within the repository.

Action: Reenter the routine name using a different value.

SSE705 THE EDITPROC NAME MUST BE SPECIFIED

Explanation: The EDITPROC name is blank.

Action: Enter a 1- to 8-character module name for the unique EDITPROC.

SSE750 COLUMN DESCRIPTION TABLE NOT FOUND, RERUN ANALYSIS

Explanation: The utility failed to respond because the column description table that is necessary to create the unique EDITPROC cannot be located within the repository.

Action: Return to the Table Selection List and Compression Statistics Panel and rerun compression analysis for the appropriate DB2 table.

SSE751 SQL ERROR OCCURRED, RERUN ANALYSIS AND REPEAT CREATE PROCESS

Explanation: The utility failed to respond because data was lost while the unique EDITPROC compression routine was being created.

Action: Return to the Table Selection List and Compression Statistics Panel rerun compression analysis for the appropriate DB2 table, and then recreate the unique EDITPROC.

Online Informational Messages

This section contains the SAMS:Compress online Informational messages. All messages issued by SAMS:Compress are preceded by a 7- or 8-character message number (SSIxxx).

SSI002 READING DB2 TABLES

Explanation: The utility is reading the tables contained within the repository before displaying the Repository Table List Panel.

SSI200Z STARTING COMPRESSION ANALYSIS PROCESS

Explanation: Self-explanatory. Normally the analysis process takes only a few minutes, depending on the parameters imposed on the analysis. However, if a time limit was imposed on the analysis, processing will be done by the end of the specified time period.

SSI200A COPE PRE-ANALYSIS: xxxx ROWS PROCESSED

Explanation: Self-explanatory. Indicates the number of rows processed through the COPE pre-analysis facility.

SSI200B COPE PRE-ANALYSIS: STARTED...

Explanation: Self-explanatory.

SSI200C ANALYZING COMPRESSION PERFORMANCE: xxxx ROWS PROCESSED

Explanation: Each of the routines specified has been evaluated for compression achieved. Currently the routine's relative performance values are being calculated.

SSI200D UPDATING SAMS:Compress REPOSITORY

Explanation: The compression analysis of the table is complete and the results of the analysis are being stored within the specified compression analysis repository.

SSI200E ANALYSIS COMPLETE, PLEASE WAIT

Explanation: Self-explanatory. The compression analysis results will be displayed on the following panel.

SSI200F COPE PRE-ANALYSIS: TOO FEW ROWS AVAILABLE (NOT COMPUTED)

Explanation: Too few rows were present in the DB2 table (or imposed by the analysis parameters) to permit the COPE facility, which requires a minimum of 2000 rows, to perform its function. Thus, no optimal routines were created.

SSI200G ANALYZING COMPRESSION PERFORMANCE: WORKING...

Explanation: Self-explanatory.

SSI200H TABLE CONTAINS ZERO ROWS

Explanation: Compression analysis was performed on a table that contained no rows. thus,no compression analysis results were calculated.

SSI210A REPOSITORY TABLES NOT FOUND FOR CREATOR AND PREFIX, CREATE NEEDED

Explanation: The repository must be created by pressing the <ENTER> key.

SSI212 variable = (xxxx)

Explanation: The utility informs you of each step during the repository table creation process.

SSI212A BUILDING SQL FOR xxxx TABLES

Explanation: The SQL statement needed to create the repository tables is being constructed.

SSI212B xxxx TABLES CREATED SUCCESSFULLY

Explanation: The repository tables have been created.

SSI216A TABLESPACE, DATABASE, STOGROUP EXIST, PRESS ENTER TO CREATE NEW TABLES

Explanation: The tablespace, database and storage group specified on the Repository Specification Panel are existing items within your system. The utility will create only the repository tables. Press <ENTER> to create the objects or change the values on the panel as needed.

SSI216B DATABASE, STOGROUP EXIST, PRESS ENTER TO CREATE NEW TABLESPACE, TABLES

Explanation: The database and storage group specified on the Repository Specification Panel are existing items within your system. The utility will create the tablespace and repository tables specified. Press <ENTER> to create the objects or change the values on the panel as needed.

SSI216C STOGROUP EXIST, PRESS ENTER TO CREATE NEW TABLESPACE, DATABASE, TABLES

Explanation: The storage group specified on the Repository Specification Panel is an existing item within your system. The utility will create the database, tablespace and repository tables specified. Press <ENTER> to create the objects or change the values on the panel as needed.

SSI216D PRESS ENTER TO CREATE NEW TABLESPACE, DATABASE, STOGROUP AND TABLES

Explanation: The utility will create the repository tables, tablespace, database and storage group specified on the Repository Specification Panel. Press <ENTER> to create the objects or change the values on the panel as needed.

SSI227A OPTIMAL PRE-ANALYSIS: (SORTING PHASE I)

Explanation: Self-explanatory.

SSI227B OPTIMAL PRE-ANALYSIS: (ANALYZING PHASE I)

Explanation: Self-explanatory.

SSI227C OPTIMAL PRE-ANALYSIS: (SORTING PHASE II)

Explanation: Self-explanatory.

SSI227D OPTIMAL PRE-ANALYSIS: (ANALYZING PHASE II)

Explanation: Self-explanatory.

SSI314A SEARCHING DB2 CATALOG FOR MATCHING TABLES, PLEASE WAIT

Explanation: The utility is preparing the list of tables for the Table Selection List and Compression Statistics Panel. To decrease the processing time, reenter the name patterns to further limit the search selection criteria.

SSI314B SEARCHING DB2 CATALOG FOR MATCHING TABLES (xxx) PROCESSED

Explanation: The utility has found the indicated number of tables to list on the Table Selection List and Compression Statistics Panel. To decrease the processing time, reenter the name patterns to further limit the search selection criteria.

SSI350A COPE HAS NOT GENERATED OPTIMAL ROUTINE INFORMATION

Explanation: The utility is unable to produce optimal routines because of the small number of rows present in the table.

SSI350B MAXIMUM TIME FOR ANALYSIS EXCEEDED, ANALYSIS INCOMPLETE

Explanation: The value specified for maximum processing time has been exceeded during the analysis.

SSI350C MAXIMUM RECORD SIZE EXCEEDS MAXIMUM ALLOWED WITH COMPRESSION

Explanation: The table being analyzed is defined with one of the following.

- LONG VARCHAR
- LONG VARGRAPHIC
- A mixture of fixed and varying length columns

These table definitions would cause the table to exceed the maximum record length if compression were implemented. The reason this would happen is that an EDITPROC requires 10 bytes of a row, thus, there are 10 bytes less for data within the row.

Action: Change the data or table definition to a maximum value which is less than the current definition. When the message is displayed, press F1 for more information about proper record length. Also, refer to the IBM manual DB2 SQL Reference for more information.

SSI701 EDITPROC CREATE IN PROGRESS, PLEASE WAIT A MOMENT

Explanation: Self-explanatory.

SSI702 **EDITPROC BUILD COMPLETE, PRESS ENTER TO
CONTINUE**

Explanation: Self-explanatory.

Index

A

- add EDITPROC 36
- additional statistics (for a given table) 16
- authorizations 13, 16, 37
- average row length
 - compressed table 27
 - size (of a table) 34
 - uncompressed table 18

B

- batch compression analysis 39
 - panel 21
 - submitting from ISPF 20
- benefits (statistical account) of compression routines 17, 25

C

- cache memory, effect on analysis 27
- command to start analysis 16
- compressed data
 - as part of row data 10
 - overview 18
- compression analysis 22
 - ISPF panels 11
 - perspective view 25
 - reported results panel 33
 - results 12
 - selecting table for analysis 16

- time considerations 20
- TSO start command 13

- compression analysis parameters
 - calculate optimal for compression versus performance 21
 - maximum number of minutes for analysis 20
- compression analysis repository 45
 - overview 10
- compression routine
 - overview 7

D

- DB2
 - control information 10

F

- features
 - SAMS:Compress 5

G

- general information
 - SAMS:Compress 3

H

- how SAMS:Compress works 7

I

ISPF compression analysis 11

N

national language compression routine

classification 8

overview 6

O

optimal routines

overview 8

R

row

compressed data, contents of 10

S

SnnFLEX1 8

SnnINTEL 8

SnnSHRVL 8

Snnccsn 8

SnnXPRES 8

software requirements 6

standard routine

overview 8

support 6

system catalog 4

T

technical support center 6

U

unique routines (EDITPROC)

relationship to optimal routines 9