User's Guide

!DB®/SMU for DB2

Version 500

Document Number TS54-5846-3

June 2000

Candle Corporation
201 North Douglas Street
El Segundo, California 90245

Contents

Preface

This *User's Guide* is designed to assist the systems programmer, database administrator, or application programmer in using !DB®/SMU for DB2. !DB/SMU is an advanced tool to support the tasks of managing and tuning DB2 table spaces and index spaces. !DB/SMU provides functionality to

- Invoke appropriate DB2 utilities to ensure the integrity of DB2 data
- Generate reports that assist in minimizing and best utilizing the DASD space required to contain the DB2 data
- Detect and assist in correcting damage to a DB2 table space or index space
- Automate the invocation of DB2 utilities for reorganization, image copy, recovery, and RUNSTATS processing

Contents of this Guide

Overview

This *User's Guide* describes the features of !DB/SMU and guides you in performing common !DB/SMU tasks. The list below describes the contents of each major section.

Content of major sections

Use the table to understand the organization and content of this guide.

User's Guide Section	Contents
What's New	This section provides information about the enhancements and changes in this release of the product.
Introducing Fundamentals	This section provides information about accessing and exiting !DB/SMU, using ISPF facilities with !DB/SMU, identifying the spaces you want to work with, and spotting trends based on high-level information provided by summary reports. It contains scenarios of common tasks you can perform using !DB/SMU.
Housekeeping and Administration	This section provides information about displaying user and system options that govern !DB/SMU processing, DB2 catalog extract history, members of the system PDS, the activity log, integrity check status, and utility profiles. It explains how to update input for user exits and run a new catalog extract.

Content of major sections (continued)

User's Guide Section	Contents
The Scan Function	This section describes the reports created by the !DB/SMU scan function and explains how to interpret the reports for table spaces and index spaces. It describes the RUNSTATS !DB/SMU gathers and the integrity checks you can implement while performing a scan.
The Space Map Function	This section describes the reports created by the !DB/SMU space map function and suggests ways you can interpret the reports both for table spaces and index spaces.
The Monitor Function	This section explains how to use the !DB/SMU monitor function to examine reports for conditions you define and trigger appropriate actions based on these conditions.
The Repair Function	This section explains how to use the !DB/SMU repair and page display functions to identify and interactively correct data base errors such as broken pages. The section walks through several examples to assist you in making modifications and creating the JCL to apply the corrections under your control and direction.
Using !DB/SMU to Initiate DB2 Utilities	This section describes the ways that you can use !DB/SMU to create necessary profiles and run selected DB2 utilities.

Content of major sections (continued)

User's Guide Section	Contents
Batch Operations	You can perform most !DB/SMU activities in batch. The Batch Operations section introduces the use of !DB/SMU in batch mode and provides instructions to create the profiles to control your batch jobs, run the jobs and create reports.
Appendixes	Appendixes to this *User's Guide* include a unit on messages issued as a result of integrity checking, and a unit that describes the sources for the data that !DB/SMU uses in its operations.
Customer Support	This section provides information for contacting Candle® Support Services and using Electronic Support. It contains the instructions, telephone numbers, and information that you need to know.

Conventions

Background

This guide uses the following conventions.

Commands All command names such as the SCANR command are shown in uppercase. This includes commands for TSO, DB2, and !DB/SMU.

Keyed data When you are given instructions to type or key data, the data to be keyed is shown in a bold font, such as **RETURN**.

Panels and figures The panels and figures in this document are representations. Actual product panels may differ.

Percent (%) sign A percent (%) sign, which supplies optimum performance, is shown in front of all CLISTs mentioned in the text. For example, a % sign precedes the reference to CLIST KTC which is written as %KTC. However, the use of the % is optional.

Variables If the data is a variable, the data is shown in an italic font, such as *databasename*. The variable name does not necessarily represent the correct number of characters allowed by the system, but instead serves as a descriptive name.

Adobe Portable Document Format

Introduction

Candle supplies documentation in the Adobe Portable Document Format (PDF). The Adobe Acrobat Reader prints PDF documents with the fonts, formatting, and graphics in the original document. To print a Candle document, do the following:

1. Specify the print options for your system. From the Acrobat Reader Menu bar, select **File > Print Setup...** and make your selections. A setting of 300 dpi is highly recommended as is duplex printing if your printer supports it.
2. To start printing, select **File > Print** on the Acrobat Reader Menu bar.
3. On the Print popup, select one of the **Print Range** options for
   - a single page
   - a range of pages
   - all of the document
4. (Optional) To fit oversize pages to the paper size currently loaded on your printer, select the **Shrink to Fit** option.

Printing problems?

Your printer ultimately determines the print quality of your output. Sometimes printing problems can occur. If you experience printing problems, potential areas to check are:

- settings for your printer and printer driver. (The dpi settings for both your driver and printer should be the same. A setting of 300 dpi is recommended.)
- the printer driver you are using. (You may need a different printer driver or the Universal Printer driver from Adobe. This free printer driver is available at www.adobe.com.)
- the halftone/graphics color adjustment for printing color on black and white printers. (Check the printer properties under **Start > Settings > Printer**. For more information, see the online help for the Acrobat Reader.)
- the amount of available memory in your printer. (Insufficient memory can cause a document or graphics to fail to print.)

For additional information on printing problems, refer to the documentation for your printer or contact your printer manufacturer.

Documentation Set

Introduction

Candle provides a complete set of documentation for !DB/SMU for DB2. Each manual in this documentation set contains a specific type of information to help you use the product.

Candle welcomes your comments and suggestions for changes or additions to the documentation set. A user comment form, located at the back of each manual, provides simple instructions for communicating with Candle's Information Development department. You can also send email to **UserDoc@candle.com**. Please include the product name, version, and book title in the subject line. To order additional manuals, contact Candle Customer Support.

The documentation listed in the following table is available for !DB/SMU.

Table 1. !DB/SMU Documentation

Doc Number	Title	Description
TS54-5846	*!DB/SMU for DB2 User's Guide*	This guide tells how to use !DB/SMU and provides a reference of functions and features.
TI51-5840	*!DB/TOOLS for DB2 Installation and Customization Guide*	This guide tells how to install !DB/SMU as well as the other !DB®/Tools for DB2 products. It also contains information on security and authorizations.

Online documentation for BookManager

In addition to the printed versions, !DB/SMU documentation is available in BookManager format. Subsequent maintenance releases of the product between major releases will contain updated BookManager documentation. See the *!DB/TOOLS for DB2 Installation and Customization Guide* for information regarding the files.

For more information about using BookManager, see the appropriate IBM documentation as follows:

IF you want to ...	**THEN see ...**
put our book on a bookshelf,	*IBM BookManager BUILD/MVS Preparing Online Books* SC38-2036
display our book,	*IBM BookManager READ/MVS Displaying Online Books* SC38-2034

Version 500 Changes

Overview

This section describes what is different between !DB/SMU for DB2 and previous versions of !DB/SMU.

Differences in this release

Primarily, the Version 500 release of !DB/SMU provides full compatibility for DB2 Version 6. However, some DB2 enhancments are not exploited by !DB/SMU Version 500.

DB2 Version 6 contains enchancements for

- defining and manipulating data objects
- conducting e-business
- improving performance and availablity of database applications
- managing the database environment
- increasing database and query capacity

This release of !DB/SMU also continues to support fully earlier releases of DB2 and is also Y2K compliant.

Changes in !DB/SMU Version 500

In addition to improved performance, this release includes changes to the Utility profiles for !DB/SMU. These changes include

- enhancements that apply CUA standards to the product panels for the Utility Profile options

  These changes affect the following Utility Profile panels:

  – option 8 for COPY 1 settings under TABLESPACE heading
  – option 12 for MODIFY settings under the TABLESPACE heading
  – option 10 for RUNSTATS settings under the TABLESPACE heading
  – option 22 for RUNSTATS settings under the INDEX heading

- a new panel for specifying parameters for binding plans and packages

  Select option 23 for the PLAN/PACKAGE settings under the the BIND/REBIND heading to display this panel. This new panel lets you specify the

- reoptimizing parameters for plans and packages
- defer preparation parameters for plans and packages

Getting Started with !DB/SMU

Chapter 1. Introduction to Using !DB/SMU

Introduction

This chapter contains an overview of the functions provided by !DB®/SMU for DB2 and the advantages these provide when you install and use !DB/SMU at your site.

Chapter Contents

What Is !DB/SMU?

Overview

This unit provides an overview of the major functions !DB/SMU provides.

Background about !DB/SMU

!DB/SMU permits you to verify the integrity of your DB2 table spaces and index spaces, perform interactive repairs to spaces with errors, and analyze space-related factors affecting DB2 performance. It provides a means to minimize and best utilize DASD space. !DB/SMU identifies exception conditions based on criteria you establish and generates DB2 utility jobstreams to automatically take corrective actions. !DB/SMU can facilitate proactive performance tuning by spotting trends and identifying hot spots.

!DB/SMU functions

This chart summarizes the activities you can perform using !DB/SMU. An overview of each of these functions and the advantages you can derive from its use are provided in the balance of this chapter.

!DB/SMU Function	Description
Scan	Generates a comprehensive set of reports and gathers statistics for each space it scans. A scan verifies data integrity for table spaces or index spaces using integrity checks you specify.
Repair	Permits you to immediately repair any space integrity errors !DB/SMU finds during a scan. You can interactively change pages and verify changes iteratively until !DB/SMU detects no errors.
Space map analysis	Reviews DB2 space map pages to produce detailed reports and graphs that give an overall picture of the condition of a given space.

!DB/SMU functions (continued)

!DB/SMU Function	Description
Monitor	Examines reports produced by !DB/SMU to identify spaces that satisfy criteria or exceed thresholds you specify. You can quickly and easily identify spaces that are candidates for reorganization or image copy.
Profiles for batch processing and utility generation	Permits you to identify spaces against which batch programs will run (run profiles) or to define exception conditions (exception profiles) that trigger batch jobs. You can predefine parameters !DB/SMU uses to generate the JCL to execute a DB2 utility (utility profiles) in either online or batch. And, you can define instructions for batch jobs that run multiple DB2 utilities and perform sequences of job steps (action profiles).
Utility generation	Based on profiles you provide, generates the complete job streams necessary to initiate many of the DB2 utilities. !DB/SMU provides interactive panels that enable you to define the parameters to take effect when !DB/SMU runs a DB2 utility.
Batch processing	Provides extensive batch facilities for automating space management. You can scan DB2 databases and generate a number of reports and save them for online review. Based on profiles you provide, !DB/SMU dynamically builds a complete and executable jobstream, tailored to your installation standards, sized to fit the dataset being processed, and ready to be submitted at your discretion.

!DB/SMU functions (continued)

!DB/SMU Function	Description
Housekeeping and administration	Permits you to display and update user and system options, such as ISPF variables, authorizations, system defaults, and user exits. You can specify the number and type of integrity checks !DB/SMU performs, and review and select utility options to use for processing.

Overview of the Scan Function

Overview

This unit provides an overview of !DB/SMU scan function and the benefits you can obtain by using it.

Elements of the scan function

The !DB/SMU scan function permits you to scan DB2 databases either online or in batch mode. You can perform the scan without having to stop the DB2 data base, its table spaces or index spaces. As a result of performing a scan, !DB/SMU generates statistics and can produce a number of reports. During the scan, !DB/SMU performs a number of integrity checks. Upon completion of the scan, you can use the results to initiate utility processing or use the !DB/SMU repair function to perform required error corrections.

Integrity checks !DB/SMU performs

The integrity checking feature !DB/SMU provides is more comprehensive than the facility provided by DB2. !DB/SMU performs the following integrity checks:

- Each page is checked for consistency against the normal format and content of a DB2 page.
- The dataset is checked to ensure that pages appear at expected intervals and that each page is of the correct type.
- Checks are made to analyze such things as free space and ID maps.
- Checks are made across pages for pointers in the index hierarchy and for both forward and backward pointers for leaf-page chains.

Online scans

During a scan, !DB/SMU reads all of the pages of the table spaces or index spaces you identify. While performing the scan, !DB/SMU analyzes the pages it has read to determine whether any data integrity problems have been found and reports the occurrence of any errors.

If !DB/SMU found any errors resulting from its comprehensive integrity checking process, you can display them for immediate correction. You can also access the series of comprehensive statistical reports !DB/SMU generates. These reports are saved and are accessible to you for future reference.

Advantages provided by the scan function

The !DB/SMU scan function provides these advantages:

- Verifies the integrity of your databases through extensive integrity checks you specify
- Creates reports either online or in batch mode
- Permits viewing of reports either online or in batch mode
- Does not require that DB2 databases be stopped
- Runs whether or not DB2 is active
- Retains generations of history based on your specification
- Provides an “intelligent scan” function that initiates scan processing for a space only if the number of updated pages exceeds a percentage you specify

Overview of the Repair Function

Overview

This unit provides an overview of the !DB/SMU repair function and the benefits you can obtain by using it.

Repairing errors discovered by a scan

If an error is detected during a scan, !DB/SMU facilitates immediate error correction using its online error correction facility and extensive online help. You can repair data by typing directly over a page dump on the screen, or you can format a page and update the format. !DB/SMU checks the new page to determine whether all errors were corrected. The process continues iteratively until !DB/SMU detects no further errors.

When no further errors are found, !DB/SMU considers the page changes certified. !DB/SMU then generates the JCL and control statements necessary to execute the DB2 REPAIR utility. You can modify the job stream !DB/SMU creates, submit it immediately, save it, or discard it.

Advantages provided by the repair function

Using the !DB/SMU repair function provides these advantages:

- Interactive repair is much faster than printing a dump, analyzing the page in error by hand, and manually creating complex control statements to run the DB2 REPAIR utility.
- Extensive online Help describes DB2 page formats and error conditions
- You can validate fixes before you apply them
- You can automatically generate the DB2 REPAIR utility
- You can practice repairs without damaging real data
- You can produce special test cases for repairs

Overview of the Space Map Analysis Function

Overview

This unit provides an overview of the !DB/SMU space map analysis function and the benefits you can obtain by using it.

Elements of the space map analysis function

DB2 maintains space map pages in all table spaces and index spaces. These space map pages contain information about how much free space is available and which pages have been updated since the last DB2 image copy.

!DB/SMU analyzes DB2 space map pages to produce numerical reports, graphs, and detailed maps of the space. The reports show the extent and distribution of free space and updated pages. An analysis of space map pages is much faster than a full database scan, but contains a lesser level of detail. Using the analysis of space map pages means you can process all your DB2 spaces nightly and keep a more current status of the state of your spaces.

Reports provided by the space map analysis function

The space map analysis function provides these reports. The reports are produced either online or in batch mode. If you produce the reports in batch mode, the output is saved to the system PDS.

Report	Purpose
Summary Report	Summary of the free space found in each space and report of the update activity for each space.
Graph of updated pages	Histogram of the distribution of updated pages throughout the table space or index space. From this graph, you can see immediately whether the updated pages are evenly distributed, clustered, or biased to some region of the table space.
Maps of updated pages	Greater detail on page updates, provided on three scales: the cylinder level, track level, and page level.

Advantages provided by the space map analysis function

Using the !DB/SMU space map analysis function provides these advantages:

- You can obtain free space, update status, and allocation information with a run time largely independent of the size of the underlying DB2 space
- Pictorial representations permit you to see at a glance the general distribution of free space in a DB2 table space, the amount of update activity in a table space, and the number and distribution of pages used in an index space,
- A “zoom” feature permits you to vary the units (pages, tracks, cylinders) of display, making it possible to use the maps effectively even for extremely large DB2 spaces

Overview of the Monitor Function

Overview

This unit provides an overview of the !DB/SMU monitor function and the benefits you can obtain by using it.

Elements of the monitor function

The purpose of the !DB/SMU monitor function is to summarize and act on the data generated by the scan and space map functions. The summary data is presented to you in an exception report. The report identifies those table spaces or index spaces that satisfy certain selection criteria or exceed specified thresholds. The exception report shows size, space, and update values for the selected spaces. You can use the information in the reports to evaluate fragmentation, the need to perform an image copy, or the need to reorganize the data space.

Using the monitor reports

The monitor function permits you to do the following with its monitor exception report:

- You can sort the list of scan and space map reports to identify the table spaces or index spaces for which a utility should be run.
- You then use commands or selects on the monitor exception report panel to perform the following functions:
  - Display report details
  - Generate the JCL to run the appropriate DB2 utility

The monitor function also creates an exceptions file. You can develop user-written programs to perform site-specific processing against this file.

Advantages provided by the monitor function

Using !DB/SMU monitor function provides these advantages:

- You can easily review and sort historical reports to isolate those spaces meeting specified conditions and take appropriate actions
- You can automatically generate and optionally execute a job for the appropriate DB2 utility or third party vendor utility

Overview of !DB/SMU Profiles

Overview

This unit discusses the way in which you can use !DB/SMU profiles to predefine or automate many !DB/SMU functions.

Types of profiles

!DB/SMU provides the types of profiles described in this chart.

Profile Type	Purpose
Run Profile	Identifies the spaces that are to be processed when you execute the batch utility. The criteria you specify in the profile define the set of spaces to be included. The set can include any combination of table spaces and index spaces. All batch scans originate with a run profile.
Utility Profile	Provides information required to execute a DB2 utility. Example variables include library names, start or restart options, device types to use, and dataset names.
Exception Profile	Defines the conditions that can trigger exception reports and optionally execute one or more DB2 utilities. Examples of exception conditions are values that exceed certain thresholds, such as a copy factor greater than 15, or meet certain conditions, such as EXIT=*exitname*.
Action Profile	Defines the actions to be taken on spaces for which exception conditions are found. You can store action profiles and reuse them. Action profiles are initiated as the result of exception conditions isolated by means of an exception profile used during a scan.

Advantages provided by !DB/SMU profiles

- Because they can be predefined, saved, and reused, profiles provide for consistent and error-free operations.
- Complex actions become testable and repeatable.
- You can automate activities based on exception conditions you specify. !DB/SMU lends itself to hands-free operations.
- Your batch processes are flexible and tailorable, yet easily managed.

Overview of Utility Generation

Overview

This unit discusses the way in which you can use !DB/SMU to generate the JCL required to execute selected DB2 utilities.

Elements of utility generation

!DB/SMU generates jobstreams to run many DB2 utilities based on the performance and use characteristics of your DB2 databases. These characteristics are defined via utility profiles. The output and work files that !DB/SMU generates are adjusted according to the size of the input datasets, thus conserving DASD space. The jobstreams are dynamically adjusted to the changing characteristics of the DB2 spaces. Submission and execution of the generated job streams remains totally under your control.

Available DB2 utilities

You can use !DB/SMU to run these DB2 utilities or their equivalent OEM versions:

- COPY (for full or incremental image copies)
- MERGECOPY
- MODIFY
- QUIESCE
- RECOVER
- REORG
- REPAIR
- RUNSTATS

Available DB2 commands

You can use !DB/SMU to execute the DB2 REBIND command for dependent plans and packages. You can also execute STOP and START DATABASE commands as needed.

Advantages provided by utility generation

!DB/SMU provides these advantages through its utility generation function:

- Automates database image copies and reorgs
- Reduces the time required for maintaining utility jobstreams
- Saves system resources
- Reduces or eliminates manual intervention to execute the utilities

Overview of Batch Processing

Overview

!DB/SMU provides a batch facility that permits you to automate many DB2 space management tasks. This unit provides summary information about the batch facility and lists its advantages.

Profiles for batch jobs

!DB/SMU provides for various types of profiles that enable you to tailor your batch jobs to fit specific needs. Profiles provide a means to define, save, and reuse different instructions and operating conditions for batch jobs. (See "Overview of !DB/SMU Profiles" on page 35 earlier in this chapter.)

Batch scans

A batch scan reads all of the pages of the spaces you identify. !DB/SMU checks each page for data integrity errors and gathers comprehensive statistics. Any errors !DB/SMU detects are printed and saved for later online repair. !DB/SMU generates a series of statistical reports that you can later review online or print.

Batch reports produced by the space map analysis function

Three reports are produced as the result of running the batch utility that executes the !DB/SMU SPACE command. These are:

- A space summary report
- Maps and graphs data, which are saved to the system PDS
- A table that contains summary results. You can use user-written SQL queries or QMF to generate reports and graphs from this data.

Batch versions of the monitor exception reports

The monitor exception reports can be printed in batch. You do so by creating a monitor run profile to define the spaces to be processed and criteria to be satisfied. A file, the monitor exceptions file, contains a record for each space that meets the criteria you specified in the monitor run profile. You can use user-written SQL queries or QMF to generate reports and graphs from the data in this table.

Advantages provided by the batch facilities

The !DB/SMU batch facilities provide the following advantages:

- You can process large amounts of data without impacting your online environment.
- You can use batch jobs to automate:
  - Collection of performance information into comprehensive reports
  - Identification of spaces that satisfy certain criteria or reach certain thresholds
  - Monitoring of space management and data performance
  - Performing a sequence of actions associated with various DB2 acitivites
  - Initiating DB2 and OEM utilities
- You can reduce errors as the result of automatic generation of utility job streams.
- You can simplify ongoing maintenance and reduce the time spent on mundane tasks.

Overview of Administrative Options

Overview

This unit describes the administrative and housekeeping options available to !DB/SMU users.

About administrative options and housekeeping

!DB/SMU provides a series of panels that permits you to customize the system. Review this chart for information.

Panel Name	Purpose
Profile Data Administration	!DB/SMU provides several housekeeping panels that are common to the !DB/Tools. Using these panels, you can specify parameters global to the !DB/Tools and to your DB2 subsystem. !DB/SMU also provides panels that permit you to describe your !DB/SMU system and user options, performance options, and integrity checking options.
User Exit Information	Use this panel to view and update information related to user exit authorizations.
Extract History	Use this panel to view generations of extract history summaries and detail.
Run New Extract	Use this panel to initiate a new extract.
System PDS	Use this panel to view the contents of the system PDS.
Actvity Log Information	Use this panel to view the contents of the !DB/SMU activity log
Allocated Profile Data	Use this panel to view the information contained in the user or the group profile.
Profile Information	Use this panel to set or to enter profile information for specific users or groups.

Prerequisite for the use of administrative and housekeeping options

To make use of many of the administrative and housekeeping options, you must have appropriate authorization.

Advantages provided by housekeeping and administrative options

The !DB/SMU administrative and housekeeping options permit you to customize your system to the needs of your site and of individual users.

Chapter 2. Introducing !DB/SMU Information

Introduction

Information about !DB/SMU is available to you in several forms.

- This guide
- Online help
- Associated publications

This chapter describes the information available to assist you in using !DB/SMU and describes how the information works together to provide the best possible support for performing !DB/SMU tasks.

Chapter Contents

Structure of This Guide

Overview

This unit shows you how you can use this guide to increase your productivity.

How the structure of this guide improves your productivity

As you become accustomed to the structure of this guide, you can improve your productivity in the following ways:

- Quickly access pertinent information you need
- Avoid viewing information that is extraneous to the task you want to perform

Sections

The major sections of this guide begin with a title page and are separated by tabs. Tabs assist you to locate these broad topics easily. The guide contains a number of major sections. You are currently using the section called "Getting Started."

Chapters

Within each section, topics are organized into chapters. A chapter contains small, related task-based units of information. The title page of each chapter introduces the information found in the chapter and provides a table of contents so you can locate specific units easily.

Units

Within each chapter are small, task-based units containing one or more pages of information. Each unit has its own title that appears as a running heading at the top of all of the pages comprising the unit. For example, the title of this unit, "Structure of This Guide," can be seen above.

Each unit starts with an overview. Reading the overview quickly tells you whether the unit contains the information you need. Thus, the amount of information you need to read is reduced.

Locating information you need

The chart shows the devices provided in this guide to help you locate the type of information you want.

Task you want to perform	Device to use
Understand the information provided by this guide	Skim the table of contents. (In this guide, the table of contents includes titles of units within each chapter so that you can determine the tasks covered in a chapter.)
Locate specific information about a task	Either • Skim the table of contents • Consult the alphabetic index at the back of the book
Find panel images for the !DB/SMU object list and housekeeping panels	Consult the sections for each of the !DB/SMU major functions: "The Scan Function" on page 151, "The Repair Function" on page 317, "The Space Map Function" on page 259, "The Monitor Function" on page 299, and "Housekeeping and Administration" on page 113.

Using !DB/SMU Online Help

Overview

This unit contains brief instructions for using the online Help available with !DB/SMU.

Categories of online Help

Online Help falls into three categories:

1. Help for !DB/SMU panels
2. Help for ISPF facilities
3. Help for messages

Background about !DB/SMU online Help

In !DB/SMU, you can use online Help to access specific information about each panel. Help is available to:

- Describe the contents of a specific field on a specific panel, if the field is a display field
- Describe the values that may be input to a given field, if the field is one that can accept input
- Provide the meaning for the selects available for a given panel
- Describe the functions of the global and local commands available for a given panel
- Describe the functions of the sorts and filters available for a given panel

Accessing online Help from a !DB/SMU menu or panel

Follow this step to access online Help.

Step	Action
1	On a !DB/SMU menu or panel, type **HELP** on the Command line and press Enter, or press the appropriate function key. **Result:** !DB/SMU displays the Help panel associated with the menu or panel you are using.

Elements of a !DB/SMU Help Panel

Overview

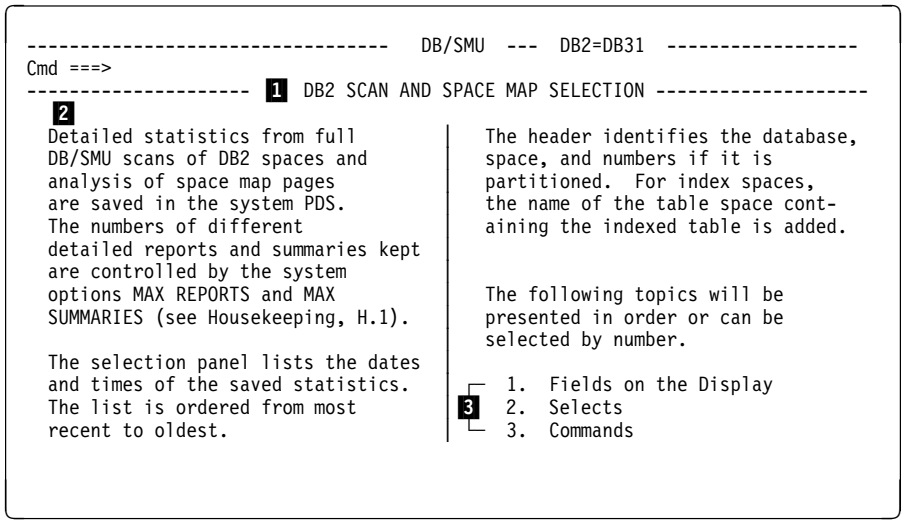
This unit illustrates the elements of a typical !DB/SMU Help panel.

Types of Help

There are two major types of Help panels in !DB/SMU. The first type provides general Help for a given panel and provides quick access pointers to the balance of the Help. The second type can be accessed by pointers from the general Help and provides detailed help for such things as fields, selects or commands on the base panel. For data entry panels, the second type lists the possible values you can enter.

General Help

This is a typical general or overview help panel.



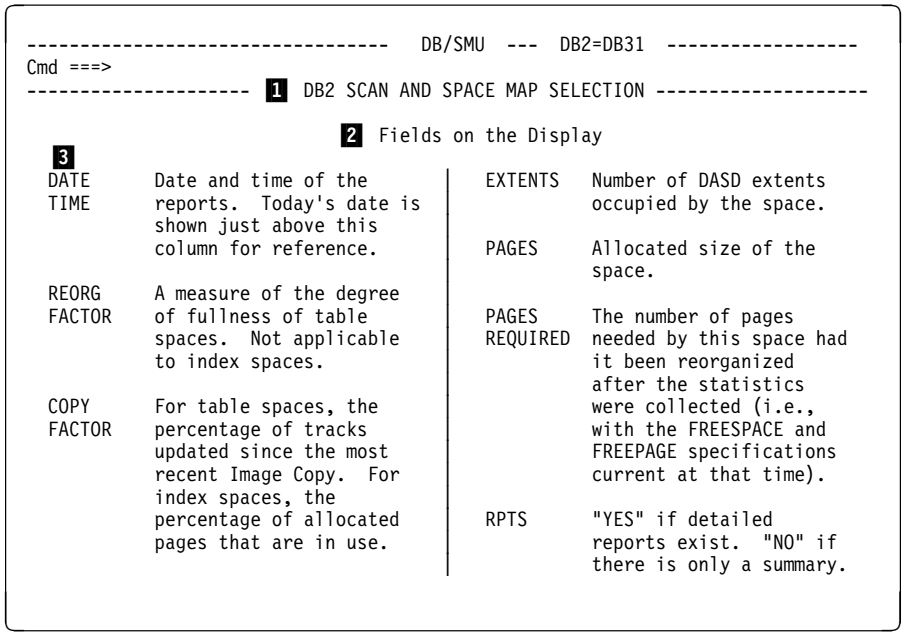
1 Title of the Help panel

2 Description of the function performed by the object panel

3 Quick access pointers to the detailed Help.

Detailed Help

This is a typical field Help panel. You use similar panels for such things as selects or commands.



```
---------------------------------  DB/SMU  ---  DB2=DB31  ------------------
Cmd ===>
---------------------- 1 DB2 SCAN AND SPACE MAP SELECTION --------------------

                            2 Fields on the Display

 3
 DATE       Date and time of the          EXTENTS   Number of DASD extents
 TIME       reports.  Today's date is               occupied by the space.
            shown just above this
            column for reference.         PAGES     Allocated size of the
                                                    space.
 REORG      A measure of the degree
 FACTOR     of fullness of table          PAGES     The number of pages
            spaces.  Not applicable       REQUIRED  needed by this space had
            to index spaces.                        it been reorganized
                                                    after the statistics
 COPY       For table spaces, the                   were collected (i.e.,
 FACTOR     percentage of tracks                    with the FREESPACE and
            updated since the most                  FREEPAGE specifications
            recent Image Copy.  For                 current at that time).
            index spaces, the
            percentage of allocated       RPTS      "YES" if detailed
            pages that are in use.                  reports exist.  "NO" if
                                                    there is only a summary.
```

1 Title of the Help panel

2 Description of the contents of this Help panel

3 List of fields and their definitions or purposes. For a field that permits data entry, the Help panel provides the values that can be entered in the field.

Help for ISPF facilities

Use this step to access Help for ISPF facilities.

Step	Action
1	On a !DB/SMU Help panel, type **HELP** on the Command line and press Enter, or press the appropriate function key. **Result:** !DB/SMU displays a general ISPF Help panel that discusses ISPF facilities.

Help for messages

If a processing error should occur when you are using !DB/SMU, an error message appears in the upper right hand corner of the panel. To get more information about the error message, review the chart below.

Step	Action
1	On a !DB/SMU menu or panel displaying the error message, type **HELP** on the Command line and press Enter, or press the appropriate function key. **Result:** !DB/SMU displays expanded information about the error condition.
2	Some messages have an additional level of Help. When the long message is displayed, you can press the appropriate function key again. If additional help is available, the system displays it.

Using Associated Publications

Overview

This unit describes the associated publications available with !DB/SMU.

Publications list

In addition to this *User's Guide*, the following associated publications can be available with !DB/SMU.

Associated publication	Purpose or use
!DB/Tools Installation Guide	Provides detailed instructions on installing !DB/SMU and other Candle&regm. database tools. It also provides information about running the !DB/SMU extract.
!DB/SMU Version Release Guide	Highlights major features of a new version or release of !DB/SMU.
Technical Bulletins (optional)	Bulletin issued to support a small programming enhancement or other small technical changes to !DB/SMU when documentation is needed but the volume of material does not warrant reissuing the *User's Guide*.

Introducing Fundamentals

Chapter 3. Accessing and Exiting !DB/SMU

Introduction

This chapter contains information on prerequisites and procedures for accessing !DB/SMU. It includes instructions for direct access and for access from OMEGAMON II for DB2.

Chapter Contents

Chapter Overview

Overview

This chapter contains instructions for accessing !DB/SMU directly. It also contains two sets of instructions for accessing !DB/SMU indirectly from OMEGAMON II for DB2.

Background about access from OMEGAMON II

You can access !DB/SMU from OMEGAMON II for DB2 in two ways:

- From the CUA interface of OMEGAMON II
- From the Primary Options Menu for OMEGAMON II for DB2 under ISPF 3.3 or higher

Determining units to use

Review the descriptions of tasks in the left column and determine which task suits your needs. Then find the page number that you need for your task.

	Prerequisites and guidelines	Direct access	OMEGAMON II for DB2 under ISPF	OMEGAMON II for DB2's CUA interface	Primary Menu overview	Exiting
	p.55	p.56	p.57	p.58	p.59	p.63
Access !DB/SMU directly	√	√				
Access !DB/SMU from the OMEGAMON II for DB2 under ISPF 3.3 or higher	√		√			
Access !DB/SMU from the OMEGAMON II for DB2 CUA interface	√			√		
Use the Primary Menu options					√	
Exit !DB/SMU						√

Prerequisites and Guidelines for Accessing !DB/SMU

Overview

This unit contains information that is common to accessing !DB/SMU, either directly from the !DB/Tools Product Selection Menu or indirectly from the OMEGAMON II for DB2 Primary Options Menu.

Prerequisites for accessing !DB/SMU

Before you use the instructions in this chapter, confirm that the !DB/Tools CLIST library is allocated in the SYSPROC concatenation. For more information, see *!DB/TOOLS for DB2 Installation and Customization Guide*. Candle Corporation recommends a TSO and batch region size of at least 4MB for any !DB/Tools product.

Information you need when accessing !DB/SMU

To access !DB/SMU, you need to know the extract ID for the database subsystem. (The extract ID is the logical DB2 subsystem ID. You identify the extract ID during installation.) For information about installation and extract procedures, see *!DB/TOOLS for DB2 Installation and Customization Guide*.

Accessing !DB/SMU Directly

Overview

Use this unit if you want to access !DB/SMU directly (without going through OMEGAMON II for DB2). Be sure you have also reviewed the unit "Prerequisites and Guidelines for Accessing !DB/SMU" on page 55.

Accessing !DB/SMU

Follow these steps to access !DB/SMU.

Step	Action
1	Type the appropriate value on the command line: • If you are on the TSO Command Processor panel (option 6 from the standard ISPF/PDF Primary Options Menu), type **%KTC** • If you are on any other ISPF panel, type: **TSO %KTC** or **EX 'hilev.CLIST(KTC)'**
2	Press Enter. **Result**: !DB/SMU displays the Candle !DB/Tools Product Selection Menu.
3	Type **2** in the Select Product ID field. If you want to use a different DB2 subsystem than the one named on the menu, type the DB2 subsystem ID in the Extract ID field.
4	Press Enter. **Result**: !DB/SMU displays the Primary Menu.

Accessing !DB/SMU from OMEGAMON II for DB2 under ISPF

Overview

Use this unit to access !DB/SMU if you are currently using OMEGAMON II for DB2 under ISPF. Be sure you have reviewed the unit "Prerequisites and Guidelines for Accessing !DB/SMU" on page 55.

Requirements for accessing !DB/SMU from OMEGAMON II for DB2

To access !DB/SMU using the instructions in this unit, you must first access OMEGAMON II for DB2 under ISPF 3.3 or higher.

Accessing !DB/SMU from OMEGAMON II for DB2 under ISPF

If you are currently using OMEGAMON II for DB2 under ISPF 3.3 or higher, follow the steps provided below to access !DB/SMU.

Step	Action
1	On the Primary Options Menu for OMEGAMON II for DB2, type **3** in the Option field.
2	Press Enter. **Result**: The system displays the Candle !DB/Tools Product Selection Menu.
3	Type **2** in the Select Product ID field. If you want to use a different DB2 subsystem than the one named on the menu, type the DB2 subsystem ID in the Extract ID field.
4	Press Enter. **Result**: !DB/SMU displays the Primary Menu.

Help if a problem occurs

If the system displays a help panel instead of the !DB/Tools Product Selection Menu when you are accessing !DB/SMU from the OMEGAMON II Primary Options Menu, see the appendix "Requirements for Access to !DB/Tools from OMEGAMON II for DB2" in the *!DB/Tools Installation and Customization Guide* for help.

Accessing !DB/SMU from the CUA Interface for OMEGAMON II for DB2

Overview

Use this unit if you want to access !DB/SMU and you are currently using the CUA interface for OMEGAMON II for DB2. Be sure you have reviewed the unit "Prerequisites and Guidelines for Accessing !DB/SMU" on page 55.

Requirements for accessing !DB/SMU from the CUA Interface

You can use the default function key (PF24) for the CUA/TSO option from OMEGAMON II for DB2 to display a pop-up menu. That menu permits you to first log onto TSO and then to access ISPF 3.3 or higher.

For general information about the CUA interface of OMEGAMON II for DB2, see the *OMEGAMON II for DB2 User's Guide*.

Accessing !DB/SMU from the CUA interface for OMEGAMON II for DB2

After you have logged onto TSO and accessed ISPF, follow these steps to access !DB/SMU.

Step	Action
1	Type the appropriate value on the command line: • If you are on the TSO Command Processor panel (option 6 from the standard ISPF/PDF Primary Options Menu), type **%KTC** • If you are on any other ISPF panel, type: **TSO %KTC** or **TSO EX 'hilev.CLIST(KTC)'**
2	Press Enter. **Result**: The system displays the Candle !DB/Tools Product Selection Menu.
3	Type **2** in the Select Product ID field.
4	Press Enter. **Result**: !DB/SMU displays the Primary Menu.

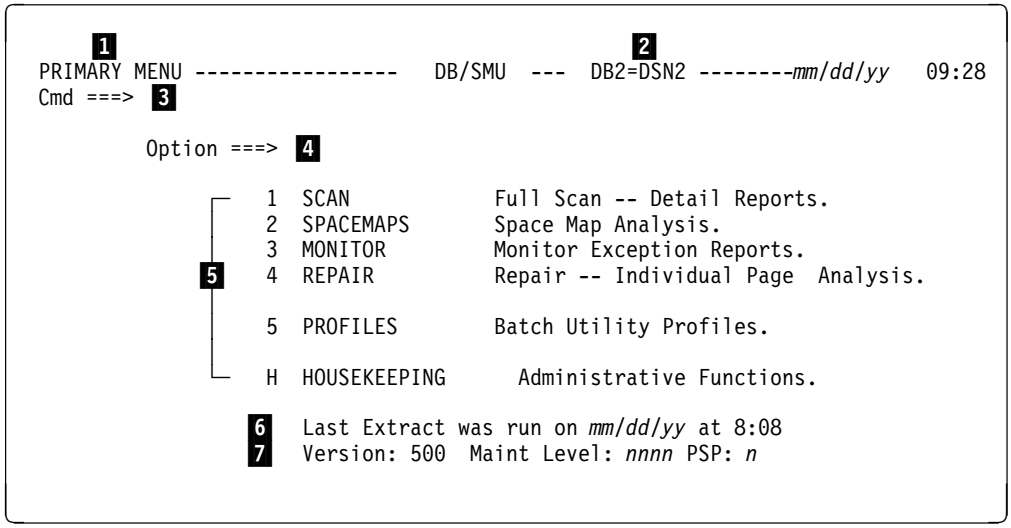
Accessing !DB/SMU Functions from the Primary Menu

Overview

This unit describes the !DB/SMU Primary Menu from which you access available functions.

Elements of the Primary Menu

This figure shows frequently used elements of the !DB/SMU Primary Menu.



1 Title of the panel

2 ID of the current DB2 subsystem

3 ISPF command line

4 Option field. Enter the number of the function to perform.

5 List of functions available with !DB/SMU

6 Date and time of the last extract

7 Version and Maintenance Level of !DB/SMU you are working with

Making a selection from the Primary Menu

!DB/SMU supports both standard menu selections and the use of multiple-level or fast path selections to access a function from the Primary Menu.

Follow these steps to make a selection from the Primary Menu.

Step	Action
1	Use one of the following selection techniques: • Type the character for the selection in the Option field. **Example**: Type **H** to select HOUSEKEEPING. • Type the fast path combination for the selection in the Option field. **Example**: Type **H.3** to select HOUSEKEEPING option 3, the option to run a new catalog extract.
2	Press Enter.

Returning to the Primary Menu

There are two methods to return to the Primary Menu. Which one you use depends upon the value you have specified in the Return=Exit field on the !DB/SMU Housekeeping panel, !DB/SMU Configuration Information.

Value for Return=Exit	Method to Return to the !DB/SMU Primary Menu
Return=Exit=N	Type **RETURN** on the command line or press the appropriate function key. You return immediately to the !DB/SMU Primary Menu without traversing any intermediate panels.

Returning to the Primary Menu (continued)

Value for Return=Exit	Method to Return to the !DB/SMU Primary Menu
Return=Exit=Y	Type **END** on the command line of each nested panel and press Enter, or press the appropriate function key, until you reach the !DB/SMU Primary Menu. (Entering **END** returns you to the next higher level panel in a nested series. Entering **RETURN** would cause you to exit !DB/SMU and return to the !DB/Tools Product Selection Menu or the OMEGAMON II Primary Options Menu.)

Effect of entering RETURN when you have made multiple selects

When you have defined RETURN to return to the !DB/SMU Primary Menu and you have made multiple selects on an object list panel, entering RETURN on a lower-level panel causes !DB/SMU to process all of your selects before executing the return.

More information about the use of RETURN and END

For additional information about these commands, see “Using END and RETURN Commands” on page 68 in the chapter “Using ISPF Facilities within !DB/SMU.”

Method for ending a session and returning to the Primary Menu

Follow the step to end a session and return to the !DB/SMU Primary Menu. (The method described for using RETURN assumes that you have defined Return=Exit=N using the !DB/SMU Housekeeping facilities.)

Step	Action
1	Perform one of the following actions: • Type **END** on the command line or the current panel and press Enter, or press the appropriate function key, as many times as necessary until you encounter the !DB/SMU Primary Menu. • Type **RETURN** on the command line of the current panel and press Enter or press the appropriate function key. • Type **=x** on the command line of the current panel and press Enter. **Result:** !DB/SMU returns you to its Primary Menu.

Background about ending a session

When you exit the Primary Menu, the session is terminated and you leave !DB/SMU.

Exiting !DB/SMU

Background about exiting !DB/SMU

You can use the !DB/SMU Change Options panel for User Options (available in Housekeeping) to control the operation of the RETURN command. You can define the RETURN command to function in one of two ways:

- Display the !DB/SMU Primary Menu
- Exit !DB/SMU

Exiting the product

The following procedure assumes that your user configuration has RETURN defined to exit !DB/SMU.

Follow the step to end a session and exit !DB/SMU.

Step	Action
1	Perform one of the following actions: • Type **END** on the command line of the current panel, or press the appropriate function key, until you have exited the Primary Menu. • Type **RETURN** on the command line of the current panel or press the appropriate function key. **Result**: The result varies based on your access method: • If you accessed !DB/SMU from OMEGAMON II for DB2, the system ends the session and returns to the Primary Options Menu for OMEGAMON II. • If you accessed !DB/SMU directly, the system exits !DB/SMU.

Chapter 4. Using ISPF Facilities within !DB/SMU

Introduction

!DB/SMU uses ISPF facilities. This chapter provides a review of information about ISPF functions, version 3.3 or higher, that are frequenty used in !DB/SMU.

Chapter Contents

Performing Some Common ISPF Operations

Overview

This unit describes the function of many of the ISPF commands you use to control your session.

Using TSO commands from a !DB/SMU panel

You can run CLISTs and EXECs and execute TSO commands from any !DB/SMU panel. Review the information in the chart for the actions to perform.

Task You Want to Perform	Action
Run a CLIST or EXEC, or execute a TSO command	On the command line, type **TSO** *commandname*, where *commandname* is the name of the CLIST or command you want to invoke.

Displaying the assignment of commands to function keys

The PFSHOW command allows you to display or suppress the display of PF keys on the bottom two lines of a panel. Displaying the PF keys can cover up the last two text lines on the panel. For example, if you use the PFSHOW command and then display a selection list panel, you may not see the last two items on the panel. Review the chart for the action to perform to display or suppress the display of PF keys.

Information You Want	Action
Assignment of function keys (those typically used in a standard ISPF environment)	Type the appropriate value on the command line: • To display the key assignments, type **PFSHOW**. • To suppress the key assignment display, type **PFSHOW OFF**. Press Enter. **Result:** !DB/SMU changes the display, including the display on the panel you are currently using.

Support for ISPF

The !DB/Tools Version 500 no longer support ISPF Version 3.1 and 3.2. All of the !DB/Tools Version 500 support ISPF Version 3.3 and above.

Displaying the panel identifier

You can turn the display of the panel identifier on and off. You may need to know the panel identifier if you contact Candle Customer Support about the behavior of a specific panel. Review the information in this chart for the action to perform.

Information You Want	Action
Panel identifier	Type the appropriate value on the command line: • To display the panel identifier, type **PANELID**. • To suppress the panel identifier, type **PANELID OFF**. Press Enter. **Result:** The system either displays or suppresses the display of the panel identifier on this panel and all subsequent panels in the session.

Creating a second ISPF session

You can create a second ISPF session and run !DB/SMU on both sessions simultaneously if your TSO USERID region size has sufficient memory. Review the instructions in the chart for the actions to perform.

Task You Want to Perform	Action
Create another ISPF display session	Type **SPLIT** on the command line and press Enter, or press the appropriate function key.
Create another ISPF display session and another !DB/SMU session	Type **SPLIT** on the command line and press Enter, or press the appropriate function key. Then start another !DB/SMU session (see “Accessing !DB/SMU Directly” on page 56).

Using END and RETURN Commands

Overview

This unit shows how you can use the END and RETURN commands to return to the !DB/SMU Primary Menu or a previous panel. It also contains information about the different ways the RETURN command can function.

Variations in the operation of the RETURN command

You can use the !DB/SMU User Options Change Options (available in Housekeeping) to define and control the operation of the RETURN command by means of your user profile. The RETURN command can function in one of two ways:

- It can display the !DB/SMU Primary Menu.
- It can exit !DB/SMU.

Returning to the !DB/SMU Primary Menu or to a previous panel

Review the information in this chart for the action to end the current panel and return to the Primary Menu or to the previous panel. (The operation of RETURN as described in the chart assumes that your user configuration has RETURN defined to display the !DB/SMU Primary Menu.)

Task You Want to Perform	Action
End operations on the current panel and access the !DB/SMU Primary Menu	Type **RETURN** on the command line and press Enter, or press the appropriate function key. **Result** !DB/SMU processes selections made on the panel. It then processes the RETURN command to display the !DB/SMU Primary Menu.
End operations on the current panel and return to the previous panel that you were using	Type **END** on the command line and press Enter, or press the appropriate function key. **Result** !DB/SMU processes the END Command. It then processes all selections you have made on the panel and displays the panel you were using previously.

Controlling the Operation of Scrolling

Overview

!DB/SMU uses standard ISPF scrolling functions. If you are not familiar with ISPF scrolling, review the information in this unit to learn how to scroll backward and forward within a document or list, move to the beginning or end of a document or list, or alter the number of lines scrolled as the result of a scroll command.

Scrolling backward or forward

Review the instructions in the chart for the actions to scroll backward and forward.

Task You Want to Perform	Action
Display a section nearer the beginning of the document or list.	Type **UP** on the command line and press Enter, or press the scroll-up function key.
Display a section farther down from the beginning of the document or list.	Type **DOWN** on the command line and press Enter, or press the scroll-up function key.
Display a section *nnn* lines closer to the beginning of the document or list.	Type **UP** *nnn* on the command line and press Enter, where *nnn* represents the number of lines you want to move to the top of the document. Example: **UP 20**
Display a section *nnn* lines closer to the end of the document or list.	Type **DOWN** *nnn* on the command line and press Enter, where *nnn* represents the number of lines you want to move to the bottom of the document. Example: **DOWN 5**

Scrolling to the beginning of a document or list

For information about scrolling to the beginning of a document or list, review the chart for two of the possible actions.

Task You Want to Perform	Action
Display a section at the beginning of the document or list.	Type **TOP** (for Maximum Scroll Up) on the command line and press Enter, or press the appropriate function key.
	Type **M** on the command line and press the function key that you assigned to the UP command.

Scrolling to the end of a document or list

When the system displays the end of a document or list, it displays a full screen rather than just the last line. It also displays a message indicating that the display is the end of the text. For information about scrolling to end of a document or list, review the information in the chart for two of the possible actions.

Task You Want to Perform	Action
Display the section that is at the end of the document or list.	Type **BOTTOM** (for Maximum Scroll Down) on the command line and press Enter, or press the appropriate function key.
	Type **M** on the command line and press the function key that you have assigned to the DOWN command.

Altering the number of lines being scrolled

Review the information in the chart for the action to perform if you want to alter the number of lines that the program each time you request it to scroll.

Task You Want to Perform	Action
Alter the number of lines scrolled in response to a scroll command.	Type the appropriate value in the scroll field. **P** (Page). The program scrolls a full page of information. **H** (Half). The program scrolls a half-page of information. **C**(Cursor). The program scrolls to the next cursor position. **<n>**. The program scrolls the specified number of lines. Press Enter. **Result:** !DB/SMU scrolls up or down the number of lines you specified. The action you take affects scrolling on all panels including the one you are currently using.

Using the HELP Command to Display Information

Overview

This unit contains information about accessing online help for !DB/SMU functions and messages, and for ISPF functions.

Displaying online help information

You can display help information for both !DB/SMU and ISPF facilities.

Review the information in this chart for the actions to display help information.

Information You Want	Action
Help for !DB/SMU functions	Type **HELP** on the command line and press Enter, or press the appropriate function key.
Help for ISPF facilities	On the panel displaying !DB/SMU help information, type **HELP** on the command line and press Enter, or press the appropriate function key.

Displaying online information about messages

This unit contains information about !DB/SMU messages that are displayed on a menu or panel.

Review this chart for the action to display information about messages.

Information You Want	Action
Information about a message that is displayed on the menu or panel	Type **HELP** on the command line and press Enter, or press the appropriate function key. **Result:** !DB/SMU displays more information about the condition.

Using the Pop-Up Command Menu

Overview

!DB/SMU makes use of a pop-up menu to display commands local to specific panels. This unit describes the use of the pop-up command menu.

Availability of the pop-up command menu

The availability of a pop-up command menu can be designated by the highlighted word **DO** in the commands list at the top of a selection list panel. Its availability can also be designated by the phrase "Enter DO for command menu" on the last line of a selection list panel.

Pop-up menus and versions of ISPF

Pop-up menus are available only if you are using ISPF version 3.0 or higher. For earlier releases, the system displays the full panel.

Elements of the pop-up command menu

The pop-up command menu can vary in content and in its location on its parent panel. It always contains the following two elements:

- list of the commands available for the parent panel
- entry field for entering your choice

Accessing the pop-up menu for commands

To use the pop-up menu that lists the commands available from the panel, follow the steps below.

Step	Action
1	On the panel, type **DO** on the command line.
2	Press Enter. **Result**: The system displays a pop-up menu of all the commands that you can use on the panel.

How to use the pop-up menu for commands

To select a command from the pop-up menu, follow the steps below.

Step	Action
1	On the pop-up menu, type the appropriate value for the command you want in the Option field.
2	Press Enter. **Result**: The system executes the process you requested.

Returning to the parent panel

If you want to return to the parent panel without selecting an option on the pop-up, type **END** in the Option field on the pop-up menu and press Enter, or press the appropriate function key.

Comparing the ISPF PRINT Command with the !DB/SMU PRNT Command

Overview

This unit compares the way in which the ISPF PRINT command differs from the !DB/SMU PRNT command. Both can be used from within a !DB/SMU session.

Background about ISPF PRINT

When you issue the ISPF PRINT command from the command line of any !DB/SMU object list panel, or press the appropriate function key, an image of the current physical screen, including any split-screen data, is written to an ISPF dataset or to the ISPF log for deferred printing.

Background about !DB/SMU PRNT

When you issue the !DB/SMU PRNT command from the command line of an object list panel, a hard copy of the panel is *immediately* spooled to a JES printer. !DB/SMU prints *the complete list of objects that comprise the display*, not just the visible contents of the screen image. That is, while you may need to scroll through the object list to make all objects visible on your screen, what !DB/SMU prints is the entire object list.

How !DB/SMU processes the PRNT command

When you issue the PRNT command, !DB/SMU:

- dynamically allocates a SYSOUT file
- writes the contents of the object list to the file
- frees the file for printing

Each time you issue the command, !DB/SMU creates a separate file and prints its contents immediately.

Format of the PRNT command

The format of the !DB/SMU PRNT command is **PRNT *class destination***, where:

- *class* is any valid SYSOUT class except *
- *destination* is any valid printer destination

When you exit !DB/SMU, ISPF saves the values you specify for class and destination as variables in your ISPF profile. !DB/SMU will use the values from your ISPF profile for all subsequent PRNT commands; you need only re-specify class or destination if you want to change its value.

Chapter 5. Identifying Spaces to Work With

Introduction

In working with !DB/SMU, you identify data spaces (table spaces and index spaces) to which you will apply various program functions. This chapter provides information about how to select candidate spaces to be used by the !DB/SMU scan, space map, and page repair functions.

Chapter Contents

Chapter Overview

Overview

This chapter describes the procedures you use to select table spaces and index spaces to be operated on by !DB/SMU.

Background about the common procedures

The procedures in this chapter are common to these functions, which you can select from the !DB/SMU Primary Menu:

1. SCAN
2. SPACEMAPS
3. MONITOR

To make the most effective use of this chapter, Candle Corporation recommends reading this chapter as you perform the actual procedures.

Prerequisites for using this chapter

Complete the following steps before reading this chapter.

1. Install the product. See the *!DB/Tools Installation and Customization Guide* for instructions on installing !DB/SMU.
2. Run the batch extract or the KTSEXSMT CLIST. (See the *!DB/Tools Installation and Customization Guide*.) These operations extract DB2 catalog information that !DB/SMU needs to identify the table spaces, tables, and index spaces in the system.
3. Read "Accessing and Exiting !DB/SMU" on page 53 and "Using ISPF Facilities within !DB/SMU" on page 65 for instructions on accessing !DB/SMU and using ISPF facilities.
4. Obtain authorization to open and read all DB2 datasets to be processed. If your facility uses an access control system, such as RACF, and you lack the appropriate authorization to open the needed DB2 datasets, RACF issues an S913 abend. !DB/SMU intercepts these abends, but you will continue to be unable to access the required DB2 datasets.

Online help and this chapter

This chapter describes procedures you perform while interacting with !DB/SMU panels. Online Help provides an additional level of detailed information about the panels and their fields.

Identifying Candidate Data Spaces

Overview

This unit describes the procedures for identifying the data spaces that the !DB/SMU scan, page repair, or space map function will operate on.

Selecting spaces in the online environment

In the online environment, !DB/SMU provides, as the first panel for the function, a Data Space Selection panel. On this panel, you specify the global criteria to limit the list of data spaces !DB/SMU will display to you for your subsequent actions.

Selecting spaces in batch

When running the function in batch, select a range of spaces that is appropriate and efficient for your needs. Processing thousands of spaces that contain huge amounts of data can increase the processing time and slow the processing rate. Consider running several jobs rather than one huge job.

For information about selecting data spaces and specifying limiting criteria in batch, see “Defining and Using Run Profiles for Batch Jobs” on page 491 and “Introduction to Handling Exception Conditions in Batch Jobs” on page 509.

Specifying criteria for data spaces

To select data spaces for scans, page repairs, or space maps, follow these steps.

Step	Action
1	On the !DB/SMU Primary Menu, choose one of the following options: SCAN SPACE MAPS REPAIR
2	Press Enter. **Result**: !DB/SMU displays a Data Space Selection panel. The heading on this panel, for example, FULL DB2 SPACE SCANS, reflects the option you selected on the !DB/SMU Primary Menu. You can review an example Data Space Selection panel in the next unit.
3	Complete all, some, or none of the fields on the Data Space Selection panel.
4	Press Enter. **Result**: !DB/SMU displays a list of data spaces meeting the criteria you defined. From this list, you can select the spaces you want to work with.

Using the Data Space Selection Panel to Create a Selection List

Overview

This unit explains how to use the Data Space Selection panel to create a selection list of data spaces for the !DB/SMU scan, page repair, or space map function.

General comments

When you specify values on a Data Space Selection panel, the following rules apply:

- For each category shown, your selection criteria can be as general or as specific as you want. You can select a large group of datasets or a few.
- If you provide no selection criteria, you automatically select *all* the table spaces and index spaces that you are currently authorized to use.
- The datasets you select using this panel are processed after you make selections on subsequent panels.

Effect of your specifications on your ISPF user profile

When you specify data space selection criteria on a Data Space Selection panel, the values you specify, with exceptions as noted, become part of your ISPF profile.

- If you exit ISPF before you log off, ISPF saves the values you entered. Each time you use !DB/SMU, these values remain in effect until you change them.
- If you log off without exiting ISPF, the values you entered are erased.

Selection criteria for data spaces

To use a Data Space Selection panel, provide a value for any of the fields, or leave all the fields blank to list all the spaces you are authorized to see. The value you enter in the fields can be a masked value. Masking is discussed in the next unit of this chapter.

```
DATA SPACE SELECTION -------   DB/SMU  ---  DB2=DSN2  ---------------------
Cmd ===>

                          FULL DB2 SPACE SCANS

       This screen allows you to specify which table spaces and
       index spaces are to be included in the select list screen.

       The name fields can be mask values using "*".

         Data Type ===>          (T) Tablespaces  (I) Indexspaces  (B) Both. 1

             Dbname ===>         Data base name. 2
        Space name ===>          Table space or index space name. 3
        Creator ID ===>          Creator authorization name. 4
      VSAM Catalog ===>          DB2 dataset high-level qualifier. 5
     Storage Group ===>          Storage group data space is assigned to. 6
       Buffer Pool ===>          Buffer pool data space is assigned to. 7
      Segment Size ===> _  _     Operator, table space segment size. 8
  Partition Number ===> _  _     Operator, partition number. 9

Spaces With Errors ===> _        (Y) Only, (N) None.  Blank for all spaces. 10
```

1 To specify the type of space you want to work with, type one of the following values:

T to include only table spaces
I to include only index spaces
B to select both table spaces and index spaces

2 Type a database name to select spaces within a specific database.

3 Type the name of a table space or index space to select a single space.

4 Type a creator ID to select spaces with a specific creator ID.

5 Type the name of a VSAM catalog to select all spaces having a specific high-level qualifier. The high-level qualifier is that one appearing in all the DB2 dataset names.

Selection criteria for data spaces (continued)

6 Type the name of a storage group to select spaces assigned to a specific storage group.

7 Type the name of a buffer pool to select spaces assigned to a specific buffer pool, for example, BP0, BP1, BP2, or BP32K.

8 Type an arithmetic operator (<, <=, =, ¬=, >=, or >) and a value to select table spaces with a certain segment size. For example, **= 0** selects nonsegmented table spaces; **>= 32** selects segmented table spaces with SEGSIZE=32 or SEGSIZE=64. (This setting is not an ISPF profile variable. It is remembered for one !DB/SMU session but not across sessions.)

9 Type an arithmetic operator (<, <=, =, ¬=, >=, or >) and a partition number for a table space or index space. For example, **>= 63** selects partitions 63 and 64 of partitioned spaces. To select nonpartitioned spaces, type **=0**. (This field is not an ISPF profile variable. It is remembered for one !DB/SMU session, but not across sessions.)

10 Type one of the following values:

Y to select table spaces or index spaces with errors
N to select table spaces or index spaces without errors
blank
to select all spaces that match the other criteria

(This field is not an ISPF profile variable. It is remembered for one !DB/SMU session, but not across sessions.)

Using Masks to Create Selection Lists

Overview

Instead of specifying actual names in the fields of the Data Space Selection panel, you can specify pattern masks. This unit provides instructions on using pattern masks.

Background about pattern masks

A pattern mask is created using special masking characters as part of your input.

Special characters used in pattern masks

This chart lists the special masking characters and explains how to use them within a string or at the end of a string.

Special Character	Used within a String	Used at the End of a String
* (asterisk)	A character exists at that position.	More characters can follow the * character.
_ (underscore)	A character exists at that position.	No character can follow the _ character. That is, when the underscore is the last character in a mask, it indicates that the character that replaces it is at the end of the name.
¬ (not)	No character exists at that position.	No character can follow the ¬ character. That is, when the not sign is the last character in a mask, it indicates that the character that precedes it is at the end of the name.

Examples of the use of pattern masks

Here is a list of names used in this example. This chart illustrates how the use of masks creates subsets of the full list.

CASPNN
CASPNN1
DASPN
DSAPN
DSQRS
DSSPN
DSSPNN
DSSPO

Names You Want	Pattern Mask to Use	Resultant Subset List
All Dbdnames that begin with DSS	DSS*	DSSPN DSSPNN DSSPO
All space names with SS as the 2nd and 3rd characters	*SS*	DSSPN DSSPNN DSSPO
All creator IDs that begin with DS and include P as the 4th character	DS*P* -or- DS_P*	DSAPN DSSPN DSSPNN DSSPO
All space names that end with NN as the 5th and 6th characters	****NN¬	CASPNN DSSPNN
All space names that have NN as the 5th and 6th characters and can have a 7th character	****NN_	CASPNN CASPNN1 DSSPNN
All VSAM catalog names that contain 5 or fewer characters	*****¬	DASPN DSAPN DSQRS DSSPN DSSPO
All space names that contain more than 5 characters	_____*	CASPNN CASPNN1 DSSPNN

Selecting Data Spaces from a Selection List

Overview

This unit describes procedures for using a selection list panel for !DB/SMU scan, page repair, and space map functions.

Background about using selection lists

As a result of providing data space selection criteria on a Data Space Selection panel, !DB/SMU displays a list of data spaces that meet the criteria you specified. The name of the selection list panel varies according to the function you chose on the !DB/SMU Primary Menu. However, the columns on each of the selection list panels are identical.

Example selection list panel

This is an example selection list panel. The list contains all of the table spaces, index spaces, or both that meet the criteria you specified on the Data Space Selection panel.

```
        1
FULL DB2 SPACE SCANS ---------   DB/SMU  ---  DB2=DSN2  ----  SPACE 1 OF 33
Cmd ===>                                                  Scroll ===> PAGE
             2                             2
             S List available Scans.      N Get a new Scan now.        10    11
3     4                    5     6          7         8           9          BUFF  SEG
SEL  TBLSPACE             PT#   DBNAME     CREATOR   CATALOG    STOGROUP   POOL  SIZE
---  ======== 12          ---   --------   --------  --------   --------   ----- ----
 _   SYSCOPY                    DSNDB06    SYSIBM    DSNCAT                BP0
 _     DSNUCH01                 DSNDB06    SYSIBM    DSNCAT     DSNCAT     BP0
 _     DSNUCX01                 DSNDB06    SYSIBM    DSNCAT     DSNCAT     BP0
 _   SYSDBASE                   DSNDB06    SYSIBM    DSNCAT                BP0
 _     DSNATX01                 DSNDB06    SYSIBM    DSNCAT     DSNCAT     BP0
 _     DSNATX02                 DSNDB06    SYSIBM    DSNCAT     DSNCAT     BP0
 _     DSNDCX01                 DSNDB06    SYSIBM    DSNCAT     DSNCAT     BP0
 _     DSNDLX01                 DSNDB06    SYSIBM    DSNCAT     DSNCAT     BP0
 _     DSNDSX01                 DSNDB06    SYSIBM    DSNCAT     DSNCAT     BP0
 _     DSNDTX01                 DSNDB06    SYSIBM    DSNCAT     DSNCAT     BP0
 _     DSNDXX01                 DSNDB06    SYSIBM    DSNCAT     DSNCAT     BP0
 _     DSNDXX02                 DSNDB06    SYSIBM    DSNCAT     DSNCAT     BP0
 _     DSNDYX01                 DSNDB06    SYSIBM    DSNCAT     DSNCAT     BP0
 _   SYSDBAUT                   DSNDB06    SYSIBM    DSNCAT                BP0
 _     DSNADH01                 DSNDB06    SYSIBM    DSNCAT     DSNCAT     BP0
 _     DSNADX01                 DSNDB06    SYSIBM    DSNCAT     DSNCAT     BP0
 _     DSNDDH01                 DSNDB06    SYSIBM    DSNCAT     DSNCAT     BP0
```

Example selection list panel (continued)

1 Title of the panel; varies with the function (scan, repair, space map analysis) you selected on the !DB/SMU Primary Panel

2 Functions you can perform on this panel. The functions you can select are described later in this unit.

3 Column in which you enter the letter representing the function you want to perform

4 Name of the table space or index space to be operated on

5 Partition number if a data space is a partitioned table space or index space

6 Name of the database to which each space belongs

7 AUTHID of the creator of each space

8 VSAM catalog name for each space

9 Storage group name associated with each space

10 Buffer pool name associated with each space

11 Segment size for each segmented table space

12 Double dashed line indicating that this is the field on which the list is currently sorted

Guidelines for using the selection list panel

The following guidelines apply to your use of a selection list panel:

- You can select multiple data spaces by entering the appropriate letter in the SEL column next to the spaces you want to process.
- You can intermix the S and N selects.
- !DB/SMU processes the selects one at a time in the order they appear on the panel.
- You can scroll the list of data spaces using the standard ISPF function keys for scrolling.
- You can print the entire list of data spaces (that is, all the data spaces that appear on the scrollable list) by entering the PRNT command on the command line.

Using Selects on a Selection List Panel

Overview

This unit describes the selects available on a selection list panel and directs you to additional information about the results obtained when you issue a select.

Operation of the select functions

Refer to the chart for information about the select functions available on the selection list panels.

Select You Use	On Selection List Panel Titled...	Performs This Action
S	Full DB2 Space Scans	Lists available scan reports See “Viewing a Report” on page 91 in this chapter for information about selecting and viewing reports.
S	DB2 Space Repair	Displays a data space for viewing or repair See the chapter “Fixing Error Pages” on page 319 for information aobut repairing pages with errors.
S	DB2 Space Map Analysis	Lists available space map analysis reports See “Viewing a Report” on page 91 in this chapter for information about selecting and viewing reports.

Operation of the select functions (continued)

Select You Use	On Selection List Panel Titled...	Performs This Action
N	Full DB2 Space Scans	Performs a new scan immediately See "Performing a New Scan or Space Map Analysis" on page 90 in this chapter for information about running new scans.
N	DB2 Space Map Analysis	Performs a new space map analysis immediately See "Performing a New Scan or Space Map Analysis" on page 90 in this chapter for information about running new space map analyses.

When the N select cannot be used

!DB/SMU does not support the use of the N select on the DB2 Space Repair panel.

Performing a New Scan or Space Map Analysis

Overview

This unit describes process for performing a new scan or new space map analysis and generating the associated report.

Process to perform a new scan or space map analysis

Follow these steps to perform a new scan or space map analysis and generate the associated report.

Step	Action
1	Access the Full DB2 Space Scans panel or the DB2 Space Map Analyses panel as described earlier in this unit.
2	On that panel, use one of these methods: • Type **N** in the SEL column next to those data spaces for which you want to perform a new scan or space map analysis; or • Type **S** in the SEL column next to those data spaces for which you want to perform a new scan or space map analysis. When !DB/SMU displays the Report Select List panel, type **S** in the Select here ===>_ field.
3	Press Enter. **Result:** !DB/SMU processes the data spaces you selected in order and performs one of these actions: • If no data integrity errors were found during the scan or space map analysis, !DB/SMU generates the appropriate report and redisplays the selection list panel. • If !DB/SMU finds errors, it displays the Scan Complete Errors panel which provides summary information about the error.

Viewing a Report

Overview

This unit describes the way in which you use the Report Select List panel to select scan reports and space map analysis reports for viewing.

Purpose of the Report Select List panel

The Report Select List panel has two major functions:

- It provides a chronological list of all of the reports of a given type (scan or space map analysis) available for a given data space.
- It provides summary statistics that permit you to see at a glance trends in dataset behavior:

  The chapter "Summary Statistics for Trend Analysis" on page 97 discusses the summary statistics and their value in trend analysis.

Availability of the Report Select List panel

The Report Select List panel is available from either the Full DB2 Space Scans selection list panel or the DB2 Space Map Analysis selection list panel.

What reports can I see?

The reports that are listed on the Report Select List panel are

- Scan reports, if you accessed this panel from the Full DB2 Space Scans selection list panel. Scan reports are described in detail in "Using the Scan Reports for Table Spaces" on page 163 and "Using the Scan Reports for Index Spaces" on page 207.
- Space map reports, if you accessed this panel from the DB2 Space Map Analysis selection list panel. Space map analysis reports are described in detail in "Performing a Space Map Analysis" on page 268.

Accessing the Report Select List panel

To access the Report Select List panel for a given data space, type an **S** in the SEL column for that data space. If you have completed a new error-free scan or space map analysis as described in the previous unit, !DB/SMU includes this new report in the Report Select List panel.

Example Report Select List panel

This is an example Report Select List panel.

```
REPORT SELECT LIST 1 ----------   DB/SMU  ---  DB2=DSN2  -------- SCAN 1 OF 27
Cmd ===>                                                  Scroll ===> PAGE
    The following  Full Space Scans  are available for Table Space SYSDBAUT 2
                        3      Data Base   DSNDB06                          ┘
   Select here     ===> _ to perform a new full space scan now.

   Select a line with    4 S Display reports   D Drop reports
                           E Erase record      R Display RUNSTATS

 Today is 11/01/99  05:33
                                 REORG    COPY                          PAGES
     SEL    DATE     TIME     FACTOR   FACTOR   EXTENTS    PAGES   REQUIRED  RPTS
     ---  --------  -----    ------   ------   -------  -------   --------  ----
      5       6       7        8        9        10        11         12      13
      _   01/10/99  05:20     87.0     82.0          5       500        441  YES
      _   01/10/99  03:33      5.0      0.0          2       200        148  YES
      _   01/08/99  03:17      5.0      0.0          2       200        112  YES
      _   01/04/99  05:04      5.0      0.0          2       200        N/A   NO
      _   12/29/98  07:13      5.0      0.0          2       200        N/A   NO
      _   12/29/98  06:38      5.0      0.0          2       200        N/A   NO
      _   12/15/98  05:17     10.0      0.0          1       100        N/A   NO
      _   11/10/98  06:29     10.0      0.0          1       100        N/A   NO
      _   11/10/98  05:09     10.0      0.0          1       100        N/A   NO
      _   11/09/98  08:13     10.0      0.0          1       100        N/A   NO
      _   11/09/98  05:50     10.0      0.0          1       100        N/A   NO
      _   11/09/98  05:47     10.0      0.0          1       100        N/A   NO
```

1 Panel title

2 Name of the DB2 space and the database to which these reports apply. If the DB2 space is an index space, !DB/SMU lists the table space that contains the indexed table. If the DB2 space is a partitioned table space or the partitioning index of a partitioned table space, !DB/SMU lists the partition number.

3 To perform a new scan or space map analysis, type **S** where indicated.

4 Functions you can perform on this panel. These select functions are described later in this unit.

5 The column where you enter the selects available on this panel.

Example Report Select List panel (continued)

6 The date on which !DB/SMU performed the scan or space map analysis. The format of the date is controlled by a user option. See "Specifying !DB/SMU Configuration Options" on page 127.

7 The time at which !DB/SMU performed the scan or space map analysis.

8 An indicator of the distribution of free space in a table space or the amount of an index that is used. See "Reorg Factor" on page 98 in the chapter "Summary Statistics for Trend Analysis" for more information.

9 An indicator of the degree of update of a table space. See "Copy Factor" on page 100 in the chapter "Summary Statistics for Trend Analysis" for more information.

10 The number of extents in the dataset. See "Number of Extents" on page 102 in the chapter "Summary Statistics for Trend Analysis" for more information.

11 The number of pages in the dataset.

12 The number of pages that would have been required by the dataset had it been reorganized at the time the report was obtained with the free space specifications in effect at that time.

13 Whether or not reports are available for the date.

YES indicates that reports are available for that date and time
NO indicates that summary data only is available. Reports are no longer available for display

Commands on a Report Select List panel

You can enter these commands on the command line of the Report Select List panel:

LOCATE The LOCATE command operates on the date in the format you have selected. If, for example, you are using the ISO format *yy/mm/dd,* and you type **L 97/3**, the list scrolls to the first report in March, 1997.

PRNT The PRNT command always sends a copy of the currently displayed list (in this case, a list of !DB/SMU full space scans) immediately to a JES printer. All lines of the list list are printed, not just those displayed on the panel.

RUNSTATS This command is available only from a full scan. Every time !DB/SMU scans a space, the values corresponding to the columns in the DB2 catalog tables updated by the DB2 RUNSTATS utility are saved in the !DB/SMU system PDS. The RUNSTATS command lists all the RUNSTATS statistics for the selected space.

Using Selects on a Report Select List Panel

Overview

This unit describes the selects available on a Report Select List Panel.

Selects available on a Report Select List panel

These selects are available on the Report Select List panel:

S Display existing reports for the selected date and !DB/SMU function (scan or space map analysis).

D Drop reports for the selected date.

E Erase a record completely.

R Display RUNSTATS (for the scan function only).

Background about historical reports

!DB/SMU maintains the number of generations of historical reports (scans or space maps) that you specify on Housekeeping's Change Options panel for System Options. Historical reports can be full-detail reports or summary reports. You specify the number of generations of each of these report types independently.

When the number of detail reports !DB/SMU maintains for you reaches the limit you defined, !DB/SMU discards the oldest report to make room for a new one. Even though !DB/SMU discards a report, it retains summary statistics from that report until the independently-specified limit for summaries is reached.

Using the drop and erase selects

Use the D (drop) and E (erase) selects to explicitly affect the retention of historical reports of either type. To use one of these selects, type it in the Sel field on the line next to the summary or detail report you want to work with.

Task You Want to Perform	Select to Use
Explicitly erase a summary statistics report	E
Explicitly erase a detail report	E or D

Background about the RUNSTATS command and the R select

You can use the RUNSTATS command or the R select only from the scan function. Their use produces different results. For information about the RUNSTATS command and R select, see the chapter “Using RUNSTATS Statistics Generated by !DB/SMU” on page 241.

Chapter 6.
Summary Statistics for Trend Analysis

Introduction

The !DB/SMU Report Select List panel provides three columns you can use to perform trend analyses. These columns are titled Reorg Factor, Copy Factor, and Extents. They help you to understand the condition of your databases and to determine whether corrective action is required. This chapter discusses these summary statistics.

Chapter Contents

Reorg Factor

Overview

This unit describes the !DB/SMU reorg factor and its use in trend analysis.

Availability of the reorg factor

The reorg factor is available as a summary statistic on the Report Select List panel. It is computed both by a full scan and by a space map analysis for a table space

Definition of the reorg factor

The reorg factor is a single value that indicates the need for database reorganization; it measures the fullness of the table space. A small value indicates that the table space has a great deal of free space; a large value indicates that the table space cannot accept many inserts without adding more extents.

What the reorg factor measures

The reorg factor measures the likelihood that DB2 will find room to insert a row in a page of the table space. There are 2 possibilities as shown in this chart:

Number of Tables in the Table Space	What the Reorg Factor Measures
Table space contains only 1 table	The reorg factor is simply the percentage of pages with no room. It is an exact indicator of the fullness of the table space and its need to be reorganized.
The table space contains multiple tables	The reorg factor is evaluated based on defined row size as follows. DB2 evaluates free space based on the defined row size of the tables in the table space. It distinguishes between the short row, the shortest of all the defined row size; the long row, the longest of all the defined row sizes; and the average row, the mean or average of the longest and shortest row sizes.

How !DB/SMU computes the reorg factor

!DB/SMU computes how many pages of the table space can accommodate table rows of each size. Pages are classified as follows:

No Room (full page)
No row can be inserted
Small Room Only short rows can be inserted
Avg Room An average row can be inserted
Large Room The longest row can be inserted

Any page with room for the longest row can also accommodate an average row or a short row. Likewise, any page with room for an average row can also hold a short row. However, the counts are not inclusive. !DB/SMU evaluates a page's free space and assigns the page to a category based on the longest row it can hold.

The reorg factor is a weighted average of the percentages of each type of page. Full pages are given full value and empty pages are given zero value, so that the final value is a reasonable indicator of fullness.

Interpreting the reorg factor

Interpreting the reorg factor requires knowledge of how a table space is used. The reorg factor is inversely related to free space in the table space. Some table spaces, especially read-only table spaces, may have little free space and yet may not require reorganization.

Reorganization of a table space is an expensive process. Tracking the table space over time helps to determine how often it needs to be reorganized. Tracking the reorg value also helps you determine appropriate options for tuning data that enable you to reduce the frequency of reorganization. For example, increasing the values for FREESPACE, FREEPAGE, or both may allow you to perform a REORG less frequently.

Copy Factor

Overview

This unit describes the copy factor and its use in trend analysis.

Availability of the copy factor

The copy factor is available as a summary statistic on the Report Select List panel. It is computed both by a full scan and by a space map analysis.

Definition of the copy factor

The copy factor is a measure of the degree of update of a table space.

How !DB/SMU determines the copy factor for table spaces

The copy factor for a table space is the percentage of total tracks of the table space modified since the last image copy. This value is chosen because it accurately indicates how long an incremental image copy will take.

Interpreting the copy factor for table spaces

The size of the copy factor can indicate whether an incremental image copy is preferable to a full image copy. Unless the value is quite small and the space itself large, the convenience of a complete image copy probably outweighs the time savings realized by doing an incremental image copy.

Using the copy factor as a trigger

You may want to use the copy factor as a trigger to initiate specific processing options when any change has occurred to the space. In order to facilitate this use of the copy factor, any change to the space will result in a non-zero copy factor. A copy factor of 0 will only occur when no changes to the space have occurred.

How !DB/SMU determines the copy factor for index spaces

The copy factor for an index space is the percentage of pages actually used in the index.

Usefulness of the copy factor for index spaces

Because of non-unique keys, it is very difficult to estimate the DASD space needed by an index until it has been created. Consequently, many index spaces are allocated more space than they need. You can realize significant DASD savings by finding those indexes that have been allocated too much space.

Interpreting the copy factor for index spaces

The copy factor is shown in the summary report lines on the Report Select List and also on the Index Keys Summary report. It is also shown for each report on the list of Monitor Exception Reports.

The Report Select List also shows the number of pages allocated to an index in the Pages column and the number of pages actually required by an index in the Pages Required column. Dividing the value in the Pages column by the value in the Pages Required column yields the copy factor.

A low percentage for the copy factor, for example, 25%, indicates that you can reduce the number of pages allocated to an index. You will need to determine the appropriate percentage for your site.

Number of Extents

Overview

This unit discusses the value of reviewing the Extents column on the Report Select List panel and the extents report you can generate to understand the number of extents in a dataset.

What does the Extents column show?

By reviewing the Extents column on the Report Select List panel, you can determine whether there is a steady growth in the number of extents in a dataset. If the number grows steadily it indicates that the dataset itself is growing. You may want to focus on such a dataset to determine its need for reorganization or reallocation, or its optimal placement on DASD.

A dataset that has many extents, especially if they are on multiple volumes or are scattered widely on a single volume can cause performance problems and should be a candidate for reorganization.

Content of the extents report

The extents report is an optional report you can generate as the result of a table space scan. repair, or space map analysis. It shows the number of extents for a given dataset. For each extent, the report lists the volume serial number of the pack the dataset is on and the tracks or cylinders each extent occupies. The two columns headed CYL HD indicate the starting and ending tracks respectively for a given extent. The CYLS column contains a value only if the dataset is allocated in cylinders.

Example extents report

This is an example extents report.

```
DSNAME   EXTENT #   VOLSER      CYL HD      CYL HD     TRKS      CYLS
         --------   ------   ---------   ---------   ------    ------
DSNH.DSNDBD.DSNDB06.SYSGPAUT.I0001.A0001
              1     VL0025      200 11      200 14        4
              2     VL0025      202  2      202  5        4
              3     VL0025      202  6      202  9        4
              4     VL0025      202 10      202 13        4
              5     VL0025      202 14      203  2        4

 TOTAL        5                                          20
```

Generating the extents report

To cause !DB/SMU to generate the extents report, include one of these statements in your KTSXSMU CLIST:

-

  ```
  ALLOC FI(XTNTMAP) DA(*) SHR
  ```

  writes the information to your terminal

-

  ```
  ALLOC FI(XTNTMAP) DA(datset_name) SHR REUSE
  ```

  writes the information to the dataset you specify

Bypassing generation of the extents report

Should you always want to bypass the generation of the extents report, you can simply exclude the **ALLOC FI(XTNTMAP)** statement from the KTSXSMU CLIST.

If you want to bypass the generation of the extents report for a specific !DB/SMU session, issue this command from the command line of a !DB/SMU panel prior to running the first scan, repair, or space map analysis:

```
TSO FREE F(XTNTMAP).
```

Chapter 7. Example Tasks

Introduction

This chapter guides you through scenarios of common tasks you can perform with !DB/SMU. An overview of each task precedes each scenario. Tasks are broken into easy-to-follow steps that include sample screen displays.

Chapter Contents

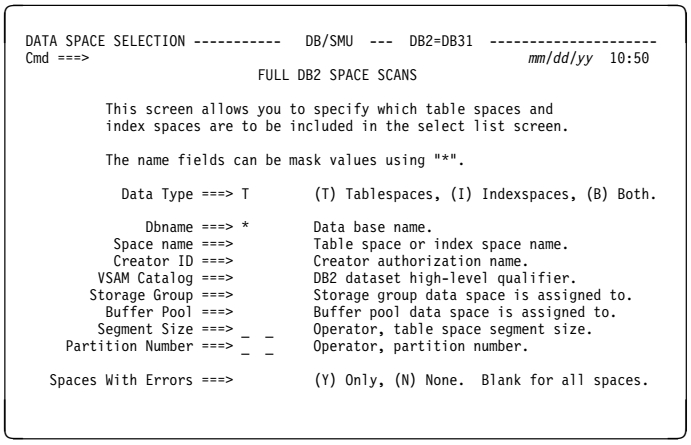
Running a Scan

Overview

In this scenario, you will:

- Select the scan function
- Enter criteria to select spaces to work with
- View a list of spaces meeting the criteria
- Run a scan

Process to view the spaces and run a scan

Step	Action
1	On the !DB/SMU Primary Menu, type **1** to select a function to work with.
2	Press Enter. **Result:** !DB/SMU displays the Data Space Selection panel titled Full DB2 Space Scans.
3	On the Data Space Selection panel, type **T** in the Data Type field to select table spaces and an asterisk (*) in the Dbname field to include all databases for which you are authorized. 
4	Press Enter. **Result:** !DB/SMU displays the Full DB2 Space Scans panel.

```
DATA SPACE SELECTION -----------   DB/SMU  ---  DB2=DB31  ----------------------
Cmd ===>                                                    mm/dd/yy  10:50
                            FULL DB2 SPACE SCANS

          This screen allows you to specify which table spaces and
          index spaces are to be included in the select list screen.

          The name fields can be mask values using "*".

            Data Type ===> T        (T) Tablespaces, (I) Indexspaces, (B) Both.

                Dbname ===> *       Data base name.
            Space name ===>         Table space or index space name.
            Creator ID ===>         Creator authorization name.
          VSAM Catalog ===>         DB2 dataset high-level qualifier.
         Storage Group ===>         Storage group data space is assigned to.
            Buffer Pool ===>        Buffer pool data space is assigned to.
           Segment Size ===> _  _   Operator, table space segment size.
      Partition Number ===> _  _    Operator, partition number.

     Spaces With Errors ===>        (Y) Only, (N) None.  Blank for all spaces.
```

Process to view the spaces and run a scan (continued)

Step	Action
5	For partition 2 of the Customer table space, type **S** in the Sel field to cause !DB/SMU to display the Report Select List panel that lists all scans for the table space or table space partition. FULL DB2 SPACE SCANS -------- DB/SMU --- DB2=DB31 ------ SPACE 2 OF 1,072 Cmd ===> Scroll ===> PAGE S List available Scans. N Get a new Scan now. SEL TBLSPACE PT# DBNAME CREATOR CATALOG STOGROUP BUFF POOL SEG SIZE _ ACT DB31COMP TSBN03 TSBN03 SGTSBN03 BP0 _ ACT DSNDB04 TSCA03 DB2V31 SYSDEFLT BP0 _ ACT DSN8DJPA TSBN05 TSBN05 DSN8GJP0 BP0 _ ACT DSN8D31A DSN8310 DB2V31 DSN8G310 BP0 _ ACTA1GRL DSNDB04 TSGM35 DB2V31 SYSDEFLT BP0 _ ALTERENQ MHLDBD TSUK72A DB2V31 SYSDEFLT BP0 _ ATEMP DBMHL ACCT TSUK TSUKSTG BP0 _ CDACTXX DBGLEN TDDB55A DB2V31 SYSDEFLT BP0 _ CDDEPTXX DBGLEN TDDB55A DB2V31 SYSDEFLT BP0 _ CDEMPPRO DBGLEN TDDB55A DB2V31 SYSDEFLT BP0 _ CDEMPXX DBGLEN TDDB55A DB2V31 SYSDEFLT BP0 _ CDPROJAC DBGLEN TDDB55A DB2V31 SYSDEFLT BP0 _ CDPROJXX DBGLEN TDDB55A DB2V31 SYSDEFLT BP0 _ CHECKTS ZETTI TSGM57 DB2V31 ZETTI BP0 _ CMPRSST1 DADD1 TDDB60 TSBN03 MMMSTO BP0 _ CUSTOMER 1 ZETTI TSGM57 DB2V31 ZETTI BP0 S CUSTOMER 2 ZETTI TSGM57 DB2V31 ZETTI BP0 _ CUSTOMER 3 ZETTI TSGM57 DB2V31 ZETTI BP0
6	Press Enter. **Result:** !DB/SMU displays the Report Select List. To run a new scan for this table space, type **S** in the Select here ===> field. REPORT SELECT LIST ---------- DB/SMU --- DB2=DB31 ---------- SCAN 1 OF 3 Cmd ===> Scroll ===> PAGE The following Full Space Scans are available for Table Space CUSTOMER Data Base ZETTI Partition 2 Select here ===> S to perform a new full space scan now. Select a line with S Display reports D Drop reports E Erase record R Display RUNSTATS Today is *mm/dd/yy* 10:50 SEL DATE TIME REORG FACTOR COPY FACTOR EXTENTS PAGES PAGES REQUIRED RPTS _ 03/07/99 09:24 16.5 100.0 1 12 3 YES _ 03/07/99 09:02 16.5 100.0 1 12 3 YES _ 03/07/99 07:53 16.5 100.0 1 12 3 YES ********** BOTTOM OF LIST **********

Process to view the spaces and run a scan (continued)

Step	Action
7	Press Enter. **Result:** !DB/SMU runs the scan, which is error free. !DB/SMU displays the Space Scan Complete panel indicating this.

```
SPACE SCAN COMPLETE ------------   DB/SMU  ---  DB2=DB31  ----------------------
Cmd ===>

    Data Base - ZETTI
  Table Space - CUSTOMER           on - mm/dd/yy


                          3 pages were scanned.

              * * * * * * * * * * * * * * * * * * * * * * * * * * *
              * * * * * * * * * * * * * * * * * * * * * * * * * * *
              * *                                                 * *
              * *  THE TABLE SPACE IS OK.  ALL TESTS PASSED.     * *
              * *          NO  ERRORS WERE DETECTED.              * *
              * *                                                 * *
              * * * * * * * * * * * * * * * * * * * * * * * * * * *
              * * * * * * * * * * * * * * * * * * * * * * * * * * *

                    Press ENTER to display statistics.

                    Press END to return to the select list.
```

Generating RUNSTATS and Updating Catalog Statistics

Overview

In this scenario, you will:

- Select a partition of a partitioned table space and view RUNSTATS for it and for the aggregate of all partitions in the table space
- Generate and view the SQL statements required to update the DB2 catalog
- Perform the update

This scenario assumes you have run a scan for the table space. For information on running a scan, see "Running a Scan" on page 106 earlier in this chapter.

Process to generate RUNSTATS and update the DB2 catalog tables

Step	Action
1	Access a Report Select List panel. If there are no scans listed for the space you want to use, run a scan as described in "Running a Scan" on page 106.
2	On the Report Select List panel, type **R** in the Sel column for the scan report you want to use.

```
REPORT SELECT LIST ----------   DB/SMU  ---  DB2=DB31  ----------- SCAN 1 OF 3
Cmd ===>                                                  Scroll ===> PAGE
    The following  Full Space Scans  are available for Table Space CUSTOMER
                                  Data Base   ZETTI     Partition    2
 Select here   ===> _ to perform a new full space scan now.

 Select a line with           S Display reports   D Drop reports
                              E Erase record      R Display RUNSTATS

 Today is 03/07/99  10:57
                                REORG    COPY                        PAGES
     SEL    DATE    TIME    FACTOR  FACTOR  EXTENTS    PAGES  REQUIRED  RPTS
     ---  --------  -----   ------  ------  -------   -------  --------  ----
      R   03/07/99  10:57    16.5     0.0        1        12         3   YES
      _   03/07/99  09:24    16.5   100.0        1        12         3   YES
      _   03/07/99  09:02    16.5   100.0        1        12         3   YES
      _   03/07/99  07:53    16.5   100.0        1        12         3   YES
     ************************ BOTTOM OF LIST ***************************
```

Process to generate RUNSTATS and update the DB2 catalog tables (continued)

Step	Action

3	!DB/SMU displays RUNSTATS for the scan. You can see the aggregate RUNSTATS for the entire table space (**1**) as well as the RUNSTATS for the partition you just scanned (**2**). You need to be aware that the aggregate values you receive are not valid unless a scan has been run for *each* of the partitions in the space, not just the one you selected in "Running a Scan." (If !DB/SMU has not gathered statistics for all partitions, the aggregate statistics will show values of -1.)
	
	`REPORT SELECT LIST ----------  DB/SMU  ---  DB2=DB31  ---------- SCAN 1 OF 3`
	`Cmd ===> GENA                                                Scroll ===> PAGE`
	`GEN  command to generate SQL UPDATEs.`
	`GENA command to include Aggregate SQL Updates.`
	`Data Base ZETTI  Table Space CUSTOMER  Part 2  on 04/07/99 at 10:57`
	`AGGREGATE 1`
	`SYSTABLESPACE`
	`NACTIVE...........312  STATSTIME... 1999/03/07 10:57:45.120000`
	`SYSTABLES  Table: CUSTOMER`
	`CARD.............. 26  NPAGES.............. 26`
	`PCTPAGES..........  8  PCTROWCOMP.........  0`
	`STATSTIME......... 1999/03/07 10:57:45.120000`
	
	`PARTITION 2`
	`SYSTABLEPART (Statistics not updatable via SQL)`
	`CARD.............. 1  FARINDREF......... 0`
	`PERCACTIVE........ 0  NEARINDREF........ 0`
	`PERCDROP.......... 0  PAGESAVE.......... 0%`
	`STATSTIME......... 1999/03/07 10:57:45.120000`
	`SYSTABSTATS  Table: CUSTOMER`
	`PART.............. 2  PCTPAGES.......... 8`
	`CARD.............. 1  NACTIVE........... 12`
	`NPAGES............ 1  PCTROWCOMP........ 0`
	`STATSTIME......... 1999/03/07 10:57:45.120000`

4	Type GENA on the command line to create SQL statements to update RUNSTATS for the entire tablespace.

5	Press Enter.
	
	Result: !DB/SMU creates the SQL statements to update the catalog and asks what you want to do.
	
	`PROCESS MEMBER -----------------  DB/SMU  ---  DB2=DB31  ----------`
	`Cmd ===>  mm/dd/yy  10:57`
	
	`RUNSTATS SQL UPDATE statements have been saved in member TDTDA9R.`
	
	`Specify next action to be performed.`
	
	`Select Action ===> 2`
	
	`1. UPDATE NOW  Invoke DB2 to execute the statements.`
	`2. EDIT  Edit saved SQL statements.`
	`3. DELETE  Delete the SQL statements member.`
	`4. PRINT  Print the SQL statements member.`
	`5. LOG  Copy member to the Log PDS.`
	
	`6. CONTINUE  Return.`

Process to generate RUNSTATS and update the DB2 catalog tables (continued)

Step	Action
6	You decide to review the update statements. Type **2** in the Select Option field to view the update statements.

7	You review the statements and now want to update the DB2 catalog. Type **END** on the command line or press the appropriate function key to return to the Process Member panel.
	
	`EDIT ---- TDKT.DBTOOLS.V31.DB31.SMU.USER(TDTDA9R) - 01.00 ---- COLUMNS 001 072`
	`COMMAND ===> END SCROLL ===> PAGE`
	`****** ***************************** TOP OF DATA ******************************`
	`000001 UPDATE SYSIBM.SYSTABLESPACE`
	`000002 SET`
	`000003 NACTIVE = 312 ,`
	`000004 STATSTIME = CURRENT TIMESTAMP`
	`000005 WHERE DBNAME = 'ZETTI'`
	`000006 AND NAME = 'CUSTOMER'`
	`000007 ;`
	`000008 UPDATE SYSIBM.SYSTABSTATS`
	`000009 SET`
	`000010 CARD = 1 ,`
	`000011 NPAGES = 1 ,`
	`000012 PCTPAGES = 8 ,`
	`000013 NACTIVE = 12 ,`
	`000014 PCTROWCOMP = 0 ,`
	`000015 STATSTIME = CURRENT TIMESTAMP`
	`000016 WHERE DBNAME = 'ZETTI'`
	`000017 AND TSNAME = 'CUSTOMER'`
	`000018 AND PARTITION = 2`
	`000019 ;`
	`000020 UPDATE SYSIBM.SYSTABLES`
	`000021 SET`
	`000022 CARD = 26 ,`
	`000023 NPAGES = 26 ,`
	`000024 PCTPAGES = 8 ,`
	`000025 PCTROWCOMP = 0 ,`
	`000026 STATSTIME = CURRENT TIMESTAMP`
	`000027 WHERE CREATOR = 'TSGM57'`
	`000028 AND NAME = 'CUSTOMER'`
	`000029 ;`
	`****** ***************************** BOTTOM OF DATA ******************************`

8	Type **1** in the Select Options field to perform the update.
	
	`PROCESS MEMBER ----------  DB/SMU  ---  DB2=DB31  ----------`
	`Cmd ===> mm/dd/yy 10:570`
	
	`RUNSTATS SQL UPDATE statements have been saved in member TDTDA9R.`
	
	`Specify next action to be performed.`
	
	`Select Action ===> 1`
	
	`1. UPDATE NOW Invoke DB2 to execute the statements.`
	`2. EDIT Edit saved SQL statements.`
	`3. DELETE Delete the SQL statements member.`
	`4. PRINT Print the SQL statements member.`
	`5. LOG Copy member to the Log PDS.`
	
	`6. CONTINUE Return.`

Process to generate RUNSTATS and update the DB2 catalog tables (continued)

Step	Action
9	Press Enter. **Result:** !DB/SMU runs the update job to update the DB2 catalog.

Housekeeping and Administration

Chapter 8. Options for Housekeeping and Administration

Introduction

This chapter contains information about the options !DB/SMU provides to support housekeeping and administration.

Chapter Contents

Background about Using the Housekeeping and Administration Options

Overview

This unit provides information about the options !DB/SMU provides that permit you to administer your system.

Accessing the housekeeping and administration functions

!DB/SMU provides a series of panels to support your use of housekeeping and administrative functions. You access these panels by selecting functions on the Administrative Functions panel.

Making changes to housekeeping and adminstration functions

Follow these steps to change a housekeeping or administrative option:

Step	Action
1	Make desired modifications to a given panel by typing new information in a field or overtyping information already in the field.
2	Press Enter. **Result:** !DB/SMU checks your entry for validity and displays the new value.

Prerequisites for changing housekeeping and administrative options

You must have appropriate authorization to update !DB/SMU housekeeping options. Review the *!DB/Tools Installation and Customization Guide* for information about authorizations.

Organization of this chapter

The units in this chapter describe the panels that comprise !DB/SMU housekeeping and administation options in the order in which the panels can be accessed from the Administrative Functions panel.

Accessing the Administrative Functions Panel

Overview

This unit provides information about accessing the Administrative Functions panel and using the options it provides.

Accessing the Administrative Functions panel

Follow these steps to access the Administrative Functions for !DB/SMU.

Step	Action
1	On the !DB/SMU Primary Menu, type **H** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the Administrative Functions panel.

Elements of the Administrative Functions panel

This is !DB/SMU Administrative Functions panel.

```
ADMINISTRATIVE FUNCTIONS-------  DB/SMU  ---  DB2=DB31  -----------------------
Cmd ===>                                                          mm/dd/yy  11:55


  Option ===>

       1      0  Profile Data Administration.
       2      1  Update Input for User Exits.
       3      2  Display DB2 Catalog Extract History.
       4      3  Run a New DB2 Catalog Extract.
       5      4  Display Members of the System PDS.
       6      5  Display Activity Log.
       7      6  Display DB/Tools profile datasets.
       8      7  Set DB/Tools profile data sets.

       9    The last Extract to TDKT.DBTOOLS.S500.DB31.SMU.SYSTEMTEM
                   was done on 05/03/99 at   3:19
```

Elements of the Administrative Functions panel (continued)

1 Displays the Profile Administration panel on which you can select the various system, database, and product profiles you want to review or update

2 Displays a panel on which you can provide input for !DB/SMU user exits

3 Displays a listing of catalog extract summary information in descending date order

4 Provides information about running a new catalog extract

5 Displays summary information about the contents of the system partitioned dataset

6 Displays a list that includes:

- Actions that updated the DB2 database
- Batch utility executions
- ABENDs

7 Displays a list of profile datasets set up earlier

8 Displays a blank panel for entering user and group profiles

9 Date and time when the most recent catalog extract was performed

Using the Profile Administration Options

Overview

This unit describes the panels !DB/SMU provides that permit you to manage your system profiles.

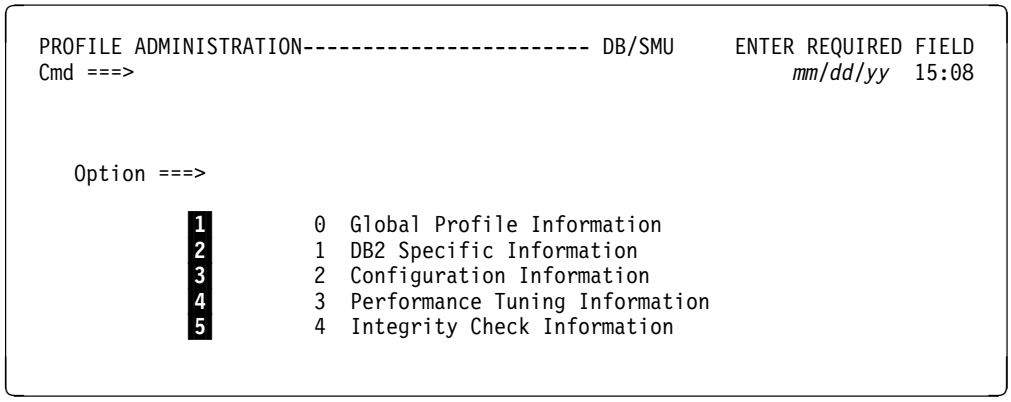
Accessing the Profile Administration panel

Follow these steps to access the Profile Administration panel:

Step	Action
1	On the Adminstrative Functions panel, type **0** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the Profile Administration panel.

Elements of the Profile Administration panel

This is !DB/SMU Profile Administration panel.



Elements of the Profile Administration panel (continued)

1 Displays the !DB/Tools Global Configuration Information panel on which you can specify configuration information common to all of the !DB/Tools

2 Displays the !DB/Tools DB2 Configuration Information panel on which you can specify configuration information for your DB2 subsystem that is common to all of the !DB/Tools

3 Displays the !DB/SMU Configuration Information panel on which you can specify configuration information specific to your !DB/SMU system

4 Displays the Performance Tuning panel on which you can specify various parameters affecting !DB/SMU performance

5 Displays the Integrity Check Status panel on which you can specify the integrity checks you want to perform during a scan

Specifying !DB/Tools Global Configuration Information

Overview

This unit describes the !DB/Tools Global Configuration Information panel.

Background about the !DB/Tools Global Configuration Information panel

The !DB/Tools Global Configuration Information panel (KTBPHOGL) displays the global !DB/Tools information for the current MVS system. You can enter new values by typing over the existing ones. The new values are written to the profile dataset that is specified on the Housekeeping panels.

Access

Follow these steps to access the !DB/Tools Global Configuration Information panel.

Step	Action
1	On !DB/SMU Profile Administration panel, type **0** in the Option field.
2	Press Enter. **Result:** !DB/SMU displays the !DB/Tools Global Configuration Information panel.

Elements of the !DB/Tools Global Configuration Information panel

This is the !DB/Tools Global Configuration Information panel.

```
 ---------------------------------- DB/Tools ----------------------------------
Cmd ===>

                  DB/Tools Global Configuration Information

CAUTION:  These are the names of your production DB/Tools data sets.
          Please be careful when changing them.

 1 CLIST  Data Set ===> TDKT.DBTOOLS.V500.CLIST
 2 CNTL   Data Set ===> TDKT.DBTOOLS.V500.CNTL
 3 LOAD   Data Set ===> TDKT.DBTOOLS.V500.LOAD
 4 DBRM   Data Set ===> TDKT.DBTOOLS.V500.DBRM
 5 MSGS   Data Set ===> TDKT.DBTOOLS.V500.MSGS
 6 PANELS Data Set ===> TDKT.DBTOOLS.V500.PANELS
 7 TABLES Data Set ===> TDKT.DBTOOLS.V500.CNTL
 8 SKELS  Data Set ===> TDKT.DBTOOLS.V500.SKELS
 9 UTIL   Data Set ===> TDKT.DBTOOLS.V500.UTIL

 10 VIO Symbolic Name       ===> VIO
 11 Temp Disk Symbolic Name ===> SYSDA
 12 Max BLKSIZE for VIO     ===> 3200
 13 SYSOUT Hold Class       ===> X
```

1 Name of the library containing the product CLISTs

2 Name of the library containing the !DB/Tools administrative control information

3 Name of the library containing the !DB/Tools LOAD modules

4 Name of the library containing the DBRMs required to BIND the product plans

5 Name of the dataset containing the product messages

6 Name of the dataset containing the product panels

7 Name of the dataset containing the product command and data tables

8 Name of the dataset containing the product skeletons

Elements of the !DB/Tools Global Configuration Information panel (continued)

9 Name of the dataset containing the product utility profiles

10 Symbolic name for virtual I/O; specific for the current MVS system

11 Symbolic name for temporary disk files; specific to a given MVS system

12 Maximum block size to use when allocating VIO datasets. Specify 0 to use the system-determined block size.

13 Your installation's hold class. If not specified, class X is assumed.

Specifying !DB/Tools DB2 Configuration Information

Overview

This unit describes the !DB/Tools DB2 Configuration Information panel

Background about the !DB/Tools DB2 Configuration Information panel

The !DB/Tools DB2 Configuration Information panel displays the !DB/Tools information for the extract ID. The extract ID is the basic unit of configuration.

Access

Follow these steps to access the !DB/Tools DB2 Configuration Information panel.

Step	Action
1	On !DB/SMU Profile Administration panel, type **1** in the Option field.
2	Press Enter. **Result:** !DB/SMU displays the !DB/Tools Global Configuration Information panel.

Elements of the !DB/Tools DB2 Configuration Information panel

This is the !DB/Tools DB2 Configuration Information panel. (!DB/SMU does not use many of the options on this panel.)

```
 ---------------------------------- DB/Tools ----------------------------------
Cmd ===>

                    DB/Tools DB2 Configuration Information

 1 Extract ID           ===> DB31             2 MVS ID           ===> SYSA
 3 Extract Description  ===> DB31 EXTRACT
 4 DB2 Subsystem ID     ===> DB31          5 DB2 Version         ===> 310
 6 DB2 Location Name    ===> DB31          7 DSNTIAUL Plan Name  ===> DSNTIB31
 8 Real Catalog Prefix  ===> SYSIBM        9 DSNTIAD  Plan Name  ===> DSNTIA31
10 Catalog Prefix       ===> SYSIBM       11 DSNZPARM Member Name ===> DSNZPARM
12 Collection Id Prefix ===> V500         13 DB/Tools Qualifier  ===> RGRAH

14 DB2 DSNLOAD Data Set         ===> TDDB2.DB2310.SDSNLOAD
15 DB2 DSNEXIT Data Set         ===> TDDB231A.DB2310.DSNEXIT
16 DB2 RUNLIB  Data Set         ===> PP.DB2.V310.RUNLIB.LOAD
17 Data Set Containing DSNHDECP ===> TDDB231A.DB2310.DSNEXIT
18 Data Set Containing DSN1COPY ===> TDDB2.DB2310.SDSNLOAD
19 Data Set Containing DSNTIAUL ===> PP.DB2.V310.RUNLIB.LOAD
20 Data Set Containing DSNTIAD  ===> PP.DB2.V310.RUNLIB.LOAD
21 Data Set Containing DSNZPARM ===> TDDB231A.DB2310.DSNEXIT
```

1 Eight character field identifying a particular extract or set of extracts on a DB2 subsystem.

2 ID of the current MVS system. The ID is for local documentation only.

3 Optional description of the contents of the extract identified by the extract ID (**1**)

4 ID of the current DB2 subsystem

5 DB2 Version of the current DB2 subsystem

6 Location name of the local DB2 subsystem (used for DB2 V2.3 or higher)

7 Name of the DSNTIAUL plan

8 The prefix of the actual DB2 catalog tables if the prefix specified for Catalog Prefix is for a shadow catalog

9 Name of the DSNTIAD plan

Elements of the !DB/Tools DB2 Configuration Information panel (continued)

10 DB2 catalog prefix

11 Name of member that contains DSNZPARMs

12 Prefix for constructing called ID for packages

!DB/SMU concatenates the CATALOG PREFIX and the COLLECTION ID PREFIX to equal the COLLECTION.

13 Name of the authorized ID (AUTHID) to be used as the qualifier of the DB/Tools tables, such as DSPSTATS and SPCSTATS

14 Name of the Dataset that contains DSNLOAD. The !DB/Tools assume that the DB2 LOADLIB (DSNLOAD) is in your system LINKLIST. See the *!DB/Tools Installation and Customization Guide* chapter entitled “Before You Begin” for additional information.

15 Name of the DSNEXIT dataset

16 Name of the DB2 RUNLIB dataset

17 Name of the dataset that contains DSNHDECP

18 Name of the dataset that contains DSN1COPY

19 Name of the dataset that contains DSNTIAUL

20 Name of the dataset that contains DSNTIAD

21 Name of the dataset that contains the DSNZPARM LOAD module

Specifying !DB/SMU Configuration Options

Overview

This unit describes the !DB/SMU Configuration Options panel.

Access

Follow these steps to access the !DB/SMU Configuration Information panel.

Step	Action
1	On !DB/SMU Profile Administration panel, type **2** in the Option field.
2	Press Enter. **Result:** !DB/SMU displays the !DB/SMU Configuration Information panel.

Elements of the !DB/SMU Configuration Information panel

This is the !DB/SMU Configuration Information panel.

```
---------------  DB/Tools Profile Information -- Product-General --------------
Cmd ===>
                        DB/SMU CONFIGURATION INFORMATION

 1 Extract PLAN Name ===> S500X     Plan Name for DB2 Extract
 2 SQL PLAN Name     ===> S500R     Plan Name for Dynamic SQL
 3 System PDS        ===> TDKT.DBTOOLS.S500.DB31.SMU.SYSTEM
 4 User PDS          ===> TDKT.DBTOOLS.S500.DB31.SMU.USER
 5 Log PDS           ===> TDKT.DBTOOLS.S500.DB31.SMU.LOG
 6 Access Exit       ===> STD       Access Control Exit (Users to Databases)
 7 Options Exit      ===> STD       Exit name for this screen
 8 Extract Exit      ===> STD       Exit to authorize users to extract Catalog
 9 Space Read Exit   ===> STD       Exit to authorize users to read data spaces
10 Max Errors        ===> 0         Max scan errors before stop  (0 = no limit)
11 Max Reports       ===> 8         Max number of reports saved per data space
12 Max Summaries     ===> 8         Max number of summaries saved per data space
13 Skip Maps Menu    ===> N         Y - Skip   N - Show Maps Menu
14 Date Format       ===> 1         1 - mm/dd/yy   2 - dd/mm/yy   3 - yy/mm/dd
15 Return = Exit?    ===> N         Y - Return exits   N - Exit to Primary Menu
16 Lines per Page    ===> 60
17 Print Class       ===> A         Print class (A-Z, 0-9)
18 Destination ID    ===>           Remote printer name or node name
19 Print to Userid   ===>           Route print to USERID at remote node
20 Special Forms ID  ===>           Printed output special forms code
```

Elements of the !DB/SMU Configuration Information panel (continued)

1 Names the DB2 plan !DB/SMU uses to run the extract

2 Names the DB2 plan !DB/SMU uses for dynamic SQL access to DB2

3 Names the system partitioned dataset. Type a fully qualified name; do not enclose it in quotes.

4 Names the user partitioned dataset. Type a fully qualified name; do not enclose it in quotes.

5 Names the log partitioned dataset. Type a fully qualified name; do not enclose it in quotes.

6 Exit you can use to control user access to and authorize the access of users to databases

7 Exit you can use to control user access to this panel

8 Exit you can use to control user access to and authorize the access of users to the catalog extract

9 Exit you can use to control a user's read access to a space

10 Controls the number of errors that are reported during a scan. When the specified number of errors is reached, !DB/SMU continues to accumulate statistics but stops reporting errors. If you plan to fix errors using the !DB/SMU repair facility, you can use the setting for Max Errors to bypass error reporting if a large number of errors occurs. In this case, you may want to consider using an alternate method to correct the errors.

11 Controls the number of reports retained in the !DB/SMU system PDS for a single table space or index space. When the specified limit is reached, a summary report replaces the oldest detail report.

!DB/SMU identifies each report with a timestamp. Thus, if you run two scans within the same minute interval, both scans can have the same timestamp. In this case, the most recent report overlays the older report. Only one report for a given timestamp is kept, since they both have the same identifier.

Elements of the !DB/SMU Configuration Information panel (continued)

12 Controls the number of summary reports kept in the !DB/SMU system PDS for a single table space or index space. When the specified limit is reached, !DB/SMU discards the oldest summary report.

13 Bypasses the summary menu that !DB/SMU displays for space map reports. The default is **N**. However, setting the value to **Y** makes viewing the space map reports more convenient, particularly from the monitor exception panels.

14 Permits you to specify the format !DB/SMU uses to display dates. This option overrides the system default.

15 Controls the use of the **RETURN** and **=X** ISPF commands and their associated function keys. When this option is set to **Y**, use of either of these commands causes you to exit !DB/SMU. When set to **N**, use of either of these commands returns you to the !DB/SMU Primary Menu.

Elements that define print options

The print options are used when you issue the PRNT command during an online session. For more information about the use of PRNT, see "Using ISPF Facilities within !DB/SMU" on page 65. You can specify the values for output class and destination when you issue the PRNT command. When you specify these values, they override the values entered on this panel. You can change the rest of the values on this panel only on the panel. These values, except lines-per-page, are ignored by the !DB/SMU batch utility.

16 Sets the number of lines to be printed per page for !DB/SMU batch reports. A value of 0 suppresses line counting. No page headings are printed except at report boundaries.

17 Names the JES output class for the output of the PRNT command. The value can be any of those from A–Z or 0–9.

18 Names the JES destination or printer ID for the output of the PRNT command. The value can be any of those allowed by your installation for JES output.

19 Names the ID to which print output is to be routed if the destination is a remote printer

20 Names the special forms to be used for PRNT command output. The value can be any of those allowed by your installation for JES output.

Specifying Performance Tuning and Miscellaneous Options

Overview

This unit describes the Administrative Functions panel that you can use to specify performance tuning and miscellaneous options.

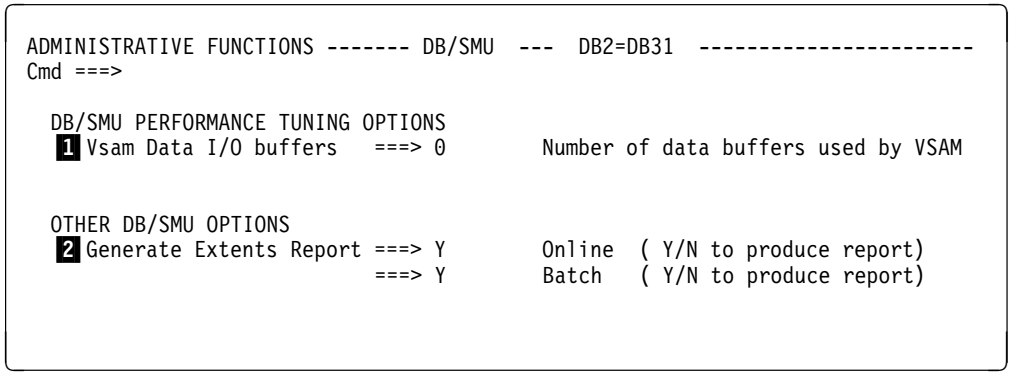
Access

Follow these steps to access the performance tuning and miscellaneous options.

Step	Action
1	On !DB/SMU Profile Administration panel, type **3** in the Option field.
2	Press Enter. **Result:** !DB/SMU displays the Administrative Functions panel used to supply performance tuning and miscellaneous option information.

Elements of performance tuning and miscellaneous options

This panel provides input for performance tuning and miscellaneous options.



1 Enter the number of VSAM I/O buffers you want to use. The value you specify is site-dependent. Generally, a larger value may improve performance.

2 Specify whether !DB/SMU is to produce an extents report for an online session, a batch session, or both

Enabling Data Integrity Checks

Overview

This unit describes the Integrity Check Status panel. A description of the various categories of checks is found in "!DB/SMU Data Integrity Checks" on page 249.

Background about the use of integrity checks

Use the Integrity Check Status panel to identify those integrity checks you want !DB/SMU to apply to a given space when performing a scan:

- If a check is ON, !DB/SMU applies the check.
- If a check is OFF, !DB/SMU suppresses the check.

Default status of a check

By default, all of the integrity checks !DB/SMU provides, with the exception of the cross-page checks (numbers 43-49 and 57-58), are set to **ON**. You must set them off to disable them.

However, if you want to use a given cross-page check, you must actively enable it.

Specifying the status of a check

Refer to this table for information about enabling or disabling a check.

Task You Want to Perform	Action to Take
Review an panel explaining a given integrity check.	Type a question mark (?) in the Sel column next to that check
Enable a specific integrity check. When enabled, !DB/SMU applies the check when scanning a space.	Type a plus sign (+) in the Sel column next to that check.
Disable a specific integrity check. When disabled, !DB/SMU ignores the check when scanning a space. ***Note:*** Before disabling a specific check, review the drawbacks to disabling a check later in this unit.	Type a minus sign (-) in the Sel column next to that check.

Accessing !DB/SMU integrity checks

Use this procedure to access the Integrity Check Status panel.

Step	Action
1	On the Profile Data Administration panel, type **4** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the Integrity Check Status panel.

Elements of the Integrity Check Status panel

This is the Integrity Check Status panel. The display shows the first 17 checks. View the remainder using ISPF scrolling to move forwards or backwards in the list.

```
INTEGRITY CHECK STATUS ------   DB/SMU  ---  DB2=DSN2  --------- CHECK 1-OF-98
Cmd ===>                                                  Scroll ===> CSR
Commands:  Locate message number; QUIT discard changes; END save changes.
Selects:   ? for detail on a check; + to enable a check; - to disable a check.
     CHECK
SEL  STATUS  MESSAGE TEXT
 1    2         3
---  ------  ------------------------------------------------------------------
 _    OFF    01  I/O ERROR FLAG ON IN HEADER
 _    ON     02  PAGE END CHARACTER MISMATCH
 _    ON     03  PAGE NUMBER MISMATCH
 _    ON     04  BROKEN PAGE FLAG IS ON
 _    ON     05  FLAGS MISMATCH -- NOT IN INDEX
 _    ON     06  FLAGS MISMATCH -- INDEX FLAG NOT ON
 _    ON     07  FLAGS MISMATCH -- HEADER PAGE FLAG INCORRECTLY ON
 _    ON     08  FLAGS MISMATCH -- HEADER PAGE FLAG NOT ON
 _    ON     09  FLAGS MISMATCH -- SPACEMAP PAGE FLAG INCORRECTLY ON
 _    ON     10  FLAGS MISMATCH -- SPACEMAP FLAG NOT ON
 _    ON     11  SPACE MAP COUNT INVALID
 _    ON     12  FREESPACE VALUE INCORRECT PAST MAP START
 _    ON     13  NUMBER OF ID MAP ENTRIES = 0 OR GT 127
 _    ON     14  LARGE HOLE FOUND NOT ON HOLE CHAIN
 _    ON     15  ERROR IN HOLE CHAIN
 _    ON     16  FREESPACE VALUE NOT = TOTAL FREESPACE
 _    ON     17  INVALID RID
```

1 Column in which to enter a plus (+) to enable a check or minus (-) to disable a check

2 Current status of a specific check

3 Number and text of the message !DB/SMU issues when the associated enabled integrity check is triggered

Drawbacks to disabling an integrity check

Disabling !DB/SMU integrity checks can have drawbacks. Few checks are independent of all others. Most checks are related to several groups of checks. Within a group, later checks are applied if all the earlier checks have been successful. If an earlier check is disabled, their dependent checks are no longer protected. The possibilities exist either that a later check can misdiagnose the situation, or that !DB/SMU can abend. For such abends, the product displays the following message:

```
99  ABEND PROCESSING PAGE
```

Note that, by default, integrity checks 43-49 and 57-58 are set to OFF. To use them, you must specifically set them ON.

Understanding User Exits

Overview

This unit discusses the purpose for !DB/SMU user exits and how you can access set-up facilities for them.

Background about the !DB/SMU exits

The primary purpose of the !DB/SMU exits is to restrict access to objects databases or spaces. These exits are set up for your site during installation. Modifying them requires high-level administrative authority. For information on how to modify the existing exits, see the *!DB/Tools Installation and Customization Guide.*

Accessing set-up facilities for user exits

Use the following procedures to access a set-up panel for the user exits.

Step	Action
1	On the Administrative Functions panel, type **1** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the current user exit information. To edit the displayed information, simply type over it.

Reviewing DB2 Catalog Extract History

Overview

This unit describes the DB2 Extract History Summary and Detail panels.

Background about the extract history summary

The DB2 Extract History panel provides a one-line summary of the results obtained each time you run an extract program against the DB2 catalog. The report lists the numbers of DB2 objects extracted. The summary lines appear in descending date order on a scrollable panel. If you performed multiple extracts on the same day, the report lists only the first of these.

Accessing the extract history summary panel

Use the following procedures to access historical summary information for your DB2 catalog extracts.

Step	Action
1	On the Administrative Functions panel, type **2** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the DB2 Extract History panel.

Elements of the DB2 Extract History panel

This is the panel displaying summary catalog extract history.

```
DB2 EXTRACT HISTORY ---------   DB/SMU  ---  DB2=D31A  -------- EXTRACT 1 OF 9
Cmd ===>                                                  Scroll ===> CSR
                   SELECT WITH  S  FOR MORE DETAIL

                    # OF       # OF       # OF       # OF       # OF   APPROX #
SEL     DATE  DATABASES  TBLSPACES     TABLES    INDEXES       KEYS   DATASETS
 1       2        3         4          5         6          7         8
--- --------  ---------  ---------  ---------  ---------  ---------  ---------
 s  11/09/99        168      1,925      1,204      2,026      2,299      3,951
 _  10/26/99        162      1,917      1,191      2,057      2,357      3,974
 _  10/15/99        151      1,881      1,142      2,034      2,299      3,915
 _  09/22/98        154        475      1,014      1,146      3,861      1,621
 _  09/20/98        155        476      1,013      1,147      3,862      1,623
 _  09/19/98        155        475      1,008      1,143      3,817      1,618
 _  09/18/98        154        475      1,008      1,143      3,817      1,618
 _  09/17/98        154        475      1,008      1,141      3,817      1,616
 _  08/29/98        150        463        984      1,129      3,757      1,592
 _  05/31/98        104        328        912      1,049      4,234      1,377
 _  05/27/98        104        328        912      1,049      4,234      1,377
 _  04/20/98         98        302        673        580      3,090        882
 ****************************** BOTTOM OF LIST *******************************
```

1 Column in which to identify the summary for which you would like additional detail. Typing **S** in this column causes !DB/SMU to display the Extract History Details panel for the extract you selected. An example Extract History Details panel can be found later in this unit.

2 Date of the extract

3 Each database counts as 1.

4 Each table space counts as 1. Each partition of a partitioned table space counts as 1.

5 Each DB2 table counts as 1.

6 Each index counts as 1. Each partition of a partitioned index counts as 1.

7 Each key field of each index counts as 1.

8 This is the sum of the values for TBLSPACES and INDEXES, since each is a separate VSAM Dataset.

Reviewing details about a specific historical extract

Follow these steps to review detailed information about a specific historical catalog extract.

Step	Action
1	On the DB2 Extract History panel, type **S** in the Sel field next to a specific date.
2	Press Enter. **Result**: !DB/SMU displays the Extract History Details panel for the summary you selected.

Elements of the Extract History Details panel

This is the Extract History Details panel.

```
EXTRACT HISTORY DETAILS------   DB/SMU  ---  DB2=DSN2  ------- VERSION 1 OF 14
Cmd ===>                                                    Scroll ===> PAGE




                                DATE = mm/dd/yy

                               20,223...CARDS IN MEMBER
                                3,951.......DATASETS
                                  168.....DATA BASES
                                1,925...TABLE SPACES
                                1,204.........TABLES
                                2,026........INDEXES
                               27,367........COLUMNS
                                2,299...........KEYS
```

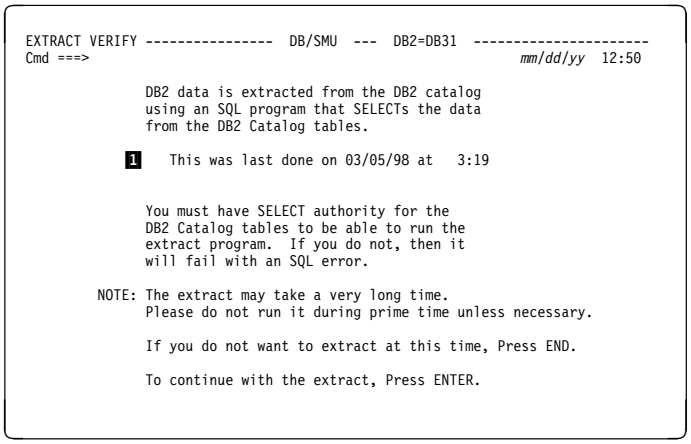
Running a New Extract

Overview

This unit illustrates the panel you use to run a new DB2 catalog extract.

Accessing the Extract Verify panel and running the extract

Use the following procedures to access the Extract Verify panel and initiate a new DB2 catalog extract.

Step	Action
1	On the Administrative Functions panel, type **3** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the Extract Verify panel. The panel shows the date and time (**1**) of the last extract and provides instructions for running a new extract.  EXTRACT VERIFY ---------- DB/SMU --- DB2=DB31 ---------- Cmd ===> *mm/dd/yy* 12:50 DB2 data is extracted from the DB2 catalog using an SQL program that SELECTs the data from the DB2 Catalog tables. **1** This was last done on 03/05/98 at 3:19 You must have SELECT authority for the DB2 Catalog tables to be able to run the extract program. If you do not, then it will fail with an SQL error. NOTE: The extract may take a very long time. Please do not run it during prime time unless necessary. If you do not want to extract at this time, Press END. To continue with the extract, Press ENTER.
3	• To run your extract immediately, press Enter. **Result:** !DB/SMU initiates a catalog extract run. • To exit the panel without running an extract, type **END** on the command line or press the appropriate function key. **Result:** !DB/SMU returns you to the Administrative Functions panel.

Viewing Information in the System PDS

Overview

This unit describes the contents of the Extract Dataset Information panel. This panel provides information about the contents of the system partitioned dataset.

Background about the Extract Dataset Information panel

The Extract Dataset Information panel provides information about where objects are stored in the system PDS. Members of the system PDS are grouped into logical members whose member name appears in encrypted form in the Member Name column. The contents of a logical member are related as, for example, all members for a DB2 database or all control members.

Accessing the details and options for the system PDS

Use the following procedures to display information about the system PDS and its options.

Step	Action
1	On the Administrative Functions panel, type **4** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the Extract Dataset Information panel.

Elements of the Extract Dataset Information panel

This is the Extract Dataset Information panel.

```
EXTRACT DATA SET INFORMATION------------------------------  ROW 1 TO 10 OF 80
Cmd ===>                                                   Scroll ==> PAGE

       The members of the Extract PDS are grouped into logical members.
    ┌─ Select with  S  to display members in the group (size is not displayed).
  1 │  Select with  R  to display members in the group (size is displayed).
    └─ Select with  E  to delete  all the members of a group.

       EXTRACT PDS NAME     TDKT.DBTOOLS.V31.DB31.SMU.SYSTEM

SEL    MEMBER NAME          MBR COUNT               TEXT
          2                    3                     4
---  ----------------       ---------       --------------------
 _   001000C400000000              26        DBD = DBHAKAN
 _   002000C400000000               9        DBD = DBKENT
 _   003000C400000000               4        DBD = DBMARIA
 _   004000C400000000              11        DBD = DBNBDB31
 _   007000C400000000               3        DBD = DHDBTDB1
 _   008000C400000000               3        DBD = DHDBTDB3
 _   009000C400000000              31        DBD = DSNDB01
 _   00A000C400000000              30        DBD = DSNDB04
 _   00B000C400000000              38        DBD = DSNDB06
 _   00C000C400000000               1        DBD = DSNDB07
 _
```

1 Selects available for use on this panel. Typing a select in the SEL column next to a member name causes !DB/SMU to display all of the individual members of the logical group the member name represents.

2 Encrypted name for a logical member in the PDS. The logical member comprises a group of related individual members.

3 Number of members in the group represented by the logical member whose encrypted name appears in the Member Name column.

4 A group identifier for the logical member whose encrypted name appears in the Member Name column.

Source of additional information

To view information about individual members of the system PDS, follow the steps in "!DB/SMU System PDS" on page 693 in the Appendix "Data Sources."

Viewing Information in the Log PDS

Overview

This unit describes the Log Activity List panel, a list of available members in the log PDS.

Use of the log PDS

!DB/SMU logs the following to the log PDS:

- Actions that updated the DB2 database
- Executions of the !DB/SMU batch utility
- ABENDs

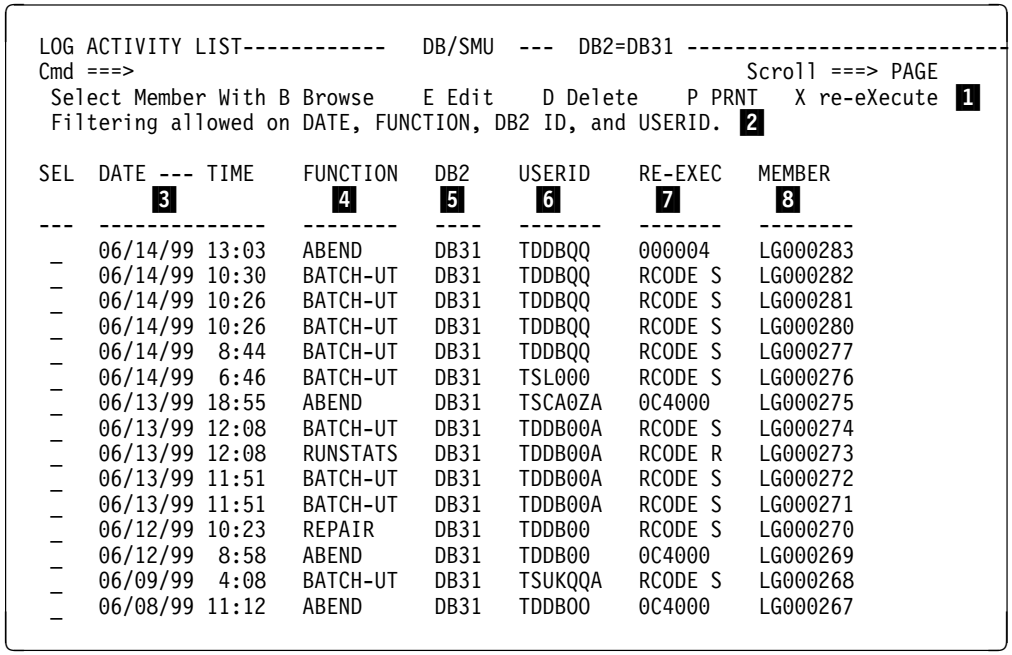
Accessing the log activity list

Use the following procedures to display information about the log PDS.

Step	Action
1	On the Administrative Functions panel, type **5** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the Log Activity List panel.

Elements of the Log Activity List panel

This is an example of the scrollable Log Activity List panel.



```
LOG ACTIVITY LIST------------   DB/SMU  ---  DB2=DB31 ----------------------------
Cmd ===>                                                    Scroll ===> PAGE
 Select Member With B Browse    E Edit    D Delete    P PRNT   X re-eXecute 1
 Filtering allowed on DATE, FUNCTION, DB2 ID, and USERID. 2

SEL  DATE --- TIME    FUNCTION   DB2    USERID    RE-EXEC   MEMBER
        3               4        5      6         7         8
---  --------------   --------   ----   -------   -------   --------
 _   06/14/99 13:03   ABEND      DB31   TDDBQQ    000004    LG000283
 _   06/14/99 10:30   BATCH-UT   DB31   TDDBQQ    RCODE S   LG000282
 _   06/14/99 10:26   BATCH-UT   DB31   TDDBQQ    RCODE S   LG000281
 _   06/14/99 10:26   BATCH-UT   DB31   TDDBQQ    RCODE S   LG000280
 _   06/14/99  8:44   BATCH-UT   DB31   TDDBQQ    RCODE S   LG000277
 _   06/14/99  6:46   BATCH-UT   DB31   TSL000    RCODE S   LG000276
 _   06/13/99 18:55   ABEND      DB31   TSCA0ZA   0C4000    LG000275
 _   06/13/99 12:08   BATCH-UT   DB31   TDDB00A   RCODE S   LG000274
 _   06/13/99 12:08   RUNSTATS   DB31   TDDB00A   RCODE R   LG000273
 _   06/13/99 11:51   BATCH-UT   DB31   TDDB00A   RCODE S   LG000272
 _   06/13/99 11:51   BATCH-UT   DB31   TDDB00A   RCODE S   LG000271
 _   06/12/99 10:23   REPAIR     DB31   TDDB00    RCODE S   LG000270
 _   06/12/99  8:58   ABEND      DB31   TDDB00    0C4000    LG000269
 _   06/09/99  4:08   BATCH-UT   DB31   TSUKQQA   RCODE S   LG000268
 _   06/08/99 11:12   ABEND      DB31   TDDBOO    0C4000    LG000267
```

1 Available selects on the log activity list panel:

- **B** Browse a member
- **D** Delete a member if you have OPTS authority
- **E** Edit a copy of the member and optionally re-execute it
- **P** Issue the PRNT command for the selected member
- **X** Reexecute the action. More information about the reexecute select can be found later in this unit.

2 Columns on which you an filter for a value you specify. Additional information about filtering can be found later in this unit.

3 Date and time the activity occurred

4 Which activity occurred

5 ID of the target DB2 system

6 User ID of the TSO user who initiated the action

7 Whether or not the action can be reexecuted

8 Member in the log PDS that contains the result of the action

About the re-execute select

If the RE-EXEC column contains an RCODE *x* value, you can re-execute the action. The *x* value indicates the contents of the member to be reexecuted:

If the value of *x* is...	Then the member contains...
S	JCL
R	SQL statements

Using filtering on the Log Activity List panel

You can filter the contents of the log activity list to limit the activities that !DB/SMU displays. Activities can be filtered by:

- Date
- Function
- DB2 ID
- User ID

Process to use filtering

Follow these steps to filter the log activity list.

Step	Action
1	On the Log Activity List panel, type **FILT** on the command line.
2	Press Enter. !DB/SMU displays the filtering pop-up menu.

```
LOG ACTIVITY LIST------------   DB/SMU  ---  DB2=DB31  ------- MEMBER 1 OF 261
Cmd ===> FILT                                          .------ LOG FILTERS ------.
 Select Member With B Browse    E Edit    D Delete     | C=>           S=> PAGE  |
 Filtering allowed on DATE, FUNCTION, DB2 ID, and U    | Comp=> OFF              |
                                                       |        E/N/O            |
SEL  DATE --- TIME    FUNCTION   DB2    USERID    R    |                         |
---  --------------   --------   ----   -------   -    | NAME VALUES             |
 _   06/14/99 13:03   ABEND      DB31   TDDBQQ    0    |                         |
 _   06/14/99 10:30   BATCH-UT   DB31   TDDBQQ    R    |    DATE=                |
 _   06/14/99 10:26   BATCH-UT   DB31   TDDBQQ    R    |    FUNC=                |
 _   06/14/99 10:26   BATCH-UT   DB31   TDDBQQ    R    |     UID=                |
 _   06/14/99  8:44   BATCH-UT   DB31   TDDBQQ    R    |     DB2=                |
 _   06/14/99  6:46   BATCH-UT   DB31   TSL000    R    | *** BOTTOM OF LIST **** |
 _   06/13/99 18:55   ABEND      DB31   TSCA0ZA   0    |                         |
 _   06/13/99 12:08   BATCH-UT   DB31   TDDB00A   R    |                         |
 _   06/13/99 12:08   RUNSTATS   DB31   TDDB00A   R    |                         |
 _   06/13/99 11:51   BATCH-UT   DB31   TDDB00A   R    |                         |
 _   06/13/99 11:51   BATCH-UT   DB31   TDDB00A   R    |                         |
 _   06/12/99 10:23   REPAIR     DB31   TDDB00    0    |                         |
 _   06/12/99  8:58   ABEND      DB31   TDDB00    0    |                         |
 _   06/09/99  4:08   BATCH-UT   DB31   TSUKQQA   R    |                         |
 _   06/08/99 11:12   ABEND      DB31   TDDB00    0    |                         |
```

Using the filtering pop-up menu

To use the filtering menu, follow these steps.

Step	Action
1	Specify a value for COMP: • Specify **E** to filter for values equal to the name value you will specify. • Specify **N** to filter for values not equal to the name value you will specify. • Specify **O** to disable filtering.
2	Specify a value for one of the following: • Date (DATE=) • Function (FUNC=) • User ID (UID=) • DB2 ID (DB2=)
3	Press Enter. **Result:** !DB/SMU filters the list on the column you specify based on the name value you enter.

Filtering examples

Example 1:

```
COMP==> N
DATE=03/13/97
```

Note: Your entry for the date and time must be in the form mm/dd/yy and hh:mm respectively. You must include any leading zeros. For example, the date must be shown as 03/13/97 rather than just 3/14/97 or you will receive erroneous results.

Example 1 result:

```
_  03/13/97 18:55 ABEND  DB31   TSCA0ZA   0C4000    LG000275
_  03/13/97 12:08 BATCH-UT DB31 TDDB00A   RCODE S   LG000274
_  03/13/97 12:08 RUNSTATS DB31 TDDB00A   RCODE R   LG000273
_  03/13/97 11:51 BATCH-UT DB31 TDDB00A   RCODE S   LG000272
_  03/13/97 11:51 BATCH-UT DB31 TDDB00A   RCODE S   LG000271
_  03/12/97 10:23 REPAIR DB31   TDDB00    RCODE S   LG000270
_  03/12/97  8:58 ABEND  DB31   TDDB00    0C4000    LG000269
_  03/09/97  4:08 BATCH-UT DB31 TSUKQQA   RCODE S   LG000268
_  03/08/97 11:12 ABEND  DB31   TDDBOO    0C4000    LG000267
```

Filtering examples (continued)

Example 2:

```
COMP==> E
FUNC=REPAIR
```

Example 2 result:

```
_   06/12/99 10:23 REPAIR DB31   TDDB00    RCODE S   LG000270
```

Viewing Information in the DB/Tools Dataset

Overview

This unit describes the contents of the DB/Tools Allocated Profile Datasets panel. The panel provides a list of allocated datasets.

Background about the DB/Tools Allocated Profile Dataset panel

The DB/Tools Allocated Profile Dataset panel provides information stored in the user and group profiles. The profile dataset contains defaults for the user with which it is associated. With authorization, users can use Houskeeping panels to change the default values in their user profile datasets.

By setting up profiles, you are placing restrictions on specified users and groups.

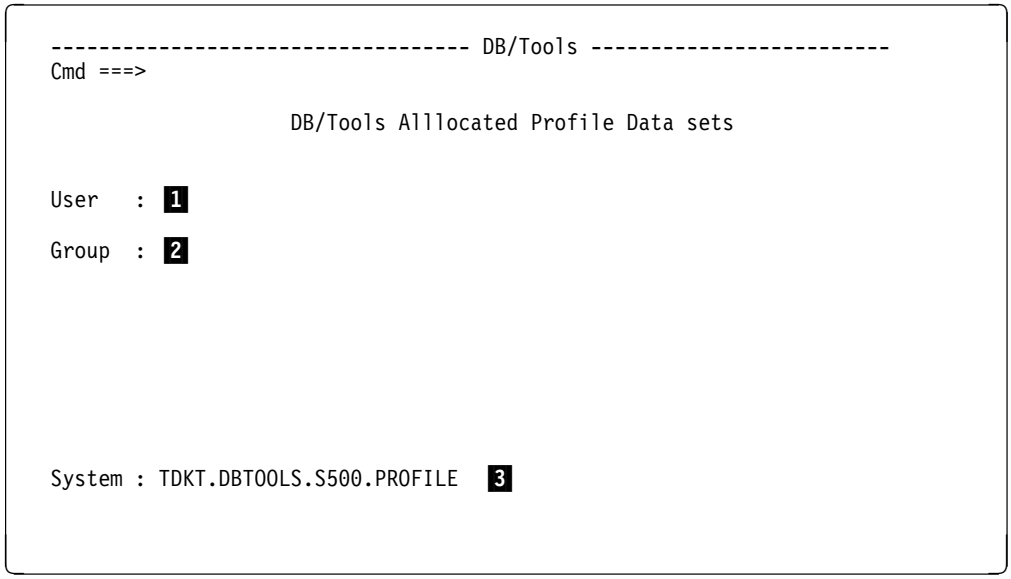
Accessing the DB/Tools Allocated Profile Dataset panel

Use the following procedures to display information about profile datasets.

Step	Action
1	On the Administrative Functions panel, type **6** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the DB/Tools Allocated Profile Datasets panel.

Elements of the DB/Tools Allocated Profile Datasets panel

This is the DB/Tools Allocated Profile Datasets panel.



1 The name of the user for whom you are setting a profile. Typing a select in the SEL column next to a member name causes !DB/SMU to display all of the individual members of the logical group the member name represents.

2 The name of the group that the user belongs to whose profile you are setting up.

3 The system where the profile resides, and from which you access and store all profile information.

Source of additional information.

To view information about individual members of the system PDS, follow the steps in "!DB/SMU System PDS" on page 693 in the Appendix "Data Sources."

Setting DB/Tools Profile Datasets

Overview

This unit describes the how to set user and group profiles on the DB/Tools Profile Information panel. Use the fields on this panel to store user defaults and values and to provide for authorization to change these defaults.

Background about the DB/Tools Information panel

The DB/Tools Information panel can be used to store specific information about a user or group. From here, you provide authorization so that a variety of functions either can or cannot be carried out from here. Default values permit users to make or to not make certain changes depending on how the profile was set up.

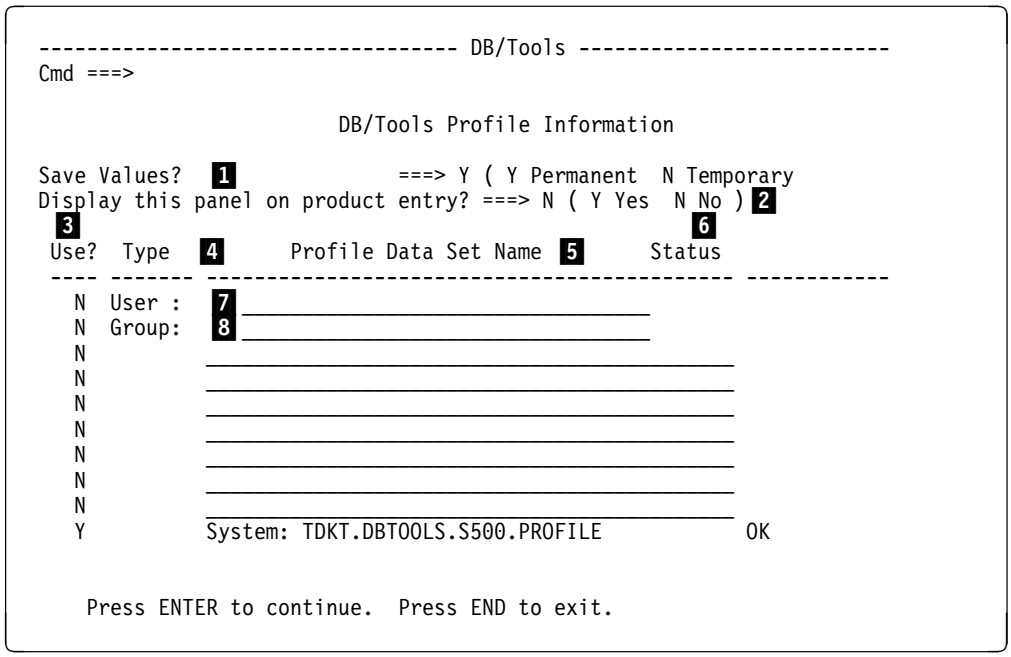
Using the DB/Tools Profile Information panel

Use the following procedures to set user and group profile datasets.

Step	Action
1	On the Administrative Functions panel, type **7** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the DB/Tools Profile Information panel.

Elements of the DB/Tools Profile Information panel

The illustration below shows the elements of the DB/Tools Profile Information panel. Use the fields to specify values for the profile datasets and the fields that display information about the profile datasets.



1 Field to specify whether the values entered are saved in your ISPF profile. Values are **Y** for permanent storage or **N** for storage to last as long as you are in this file. Default is **Y**

2 Field to specify whether the product displays the panel when you access the product. Values are **N** or **Y.** Default is **N**

3 Field to specify whether the product uses the profile dataset specified in the field. Default is **N**

4 Display field indicating the type of profile dataset

5 Fields for entering the names of the profile datasets

6 Display field indicating the status of the profile dataset specified in the field

7 Display field indicating the name of the system profile dataset. The system profile dataset was specified when the product was first installed

8 Display field indicating the name of the group

The Scan Function

Chapter 9.
Using the Scan Function

Introduction

This chapter contains information on using the !DB/SMU scan function.

Chapter Contents

Chapter Overview

Prerequisites for using this chapter

Before running the scan function, do the following:

- Review the information on selecting spaces in “Identifying Spaces to Work With” on page 77
- Ensure that you have appropriate authority for !DB/SMU
- Ensure that you have appropriate authority for dataset access required by a control system such as RACF

Organization of information in this chapter

This chapter describes the scan function and covers procedures for performing a scan.

Online help and this chapter

This chapter contains descriptions of the online panels you use to perform a scan and the procedures you follow. Online help provides an additional level of detailed information about the panels and their fields.

Overview of the Scan Function

Overview

This unit describes the way in which the scan function operates and lists the output of a scan.

How a scan works

The scan function verifies the integrity of a table space or index space and can be performed either online or in batch mode. As !DB/SMU scans a table space or index space, it performs the following actions:

- Reads the entire space
- Checks each page for validity
- Checks any pointers that might exist in the table space or index space
- Generates a series of statistical reports for the table space or index space. These reports are stored in a PDS member for review at a later time. A batch scan also produces a printed report.
- Calculates !DB/SMU RUNSTATS statistics for the table space or index space. A description of the RUNSTATS produced by a scan can be found in "Using RUNSTATS Statistics Generated by !DB/SMU" on page 241 later in this section.
- Provides an immediate opportunity to repair any data space integrity errors that are found during the scan. A description of the integrity checks !DB/SMU performs is found later in this section in the chapter "!DB/SMU Data Integrity Checks" on page 249.

The next unit lists the reports produced by a scan.

Caution

You can run a full database scan without stopping the space. However, this can lead to !DB/SMU issuing errors that are false alarms or erroneous. To learn how to determine whether an error is erroneous and how to avoid erroneous errors, see "Problem Resolution during a Scan" on page 160 later in this chapter.

Reports produced by a scan

The !DB/SMU full space scan produces these statistical reports for table spaces and index spaces:

- Reports for all table spaces
  - Pages Analysis
  - Update Graph
  - Pointer Analysis
  - Free Space Analysis and Graphs
  - DASD Space Analysis
  - Table Rows Summary
  - Table Space Usage
- Reports for segmented table spaces
  - Segment Statistics
  - Segment Map
- Reports for compressed table spaces
  - Dictionary Report
  - Compressed Rows
  - Compressed Space Usage
- Reports for index spaces
  - Keys Summary
  - Distribution of Pages with Room for Keys
  - Distribution of Records per Key
  - Top Ten Keys List
  - Pages Analysis
  - Usage Graph
  - Free Space Graph
  - Index Subpages Analysis (for type 1 indexes)
  - Index Leaf Pages Analysis (for type 2 indexes)
  - Index Levels Summary

Performance of a scan while using !DB/SMU online

When you initiate an online scan of all DB2 pages, the DB2 dataset is allocated, opened, read, closed, and deallocated. A scan normally runs about 1–2 cylinders per second. When scanning large spaces, you should consider using !DB/SMU in batch.

Performing a Scan

Overview

This unit contains procedures for performing an online scan.

Beginning a scan

Review the "Prerequisites for using this chapter" on page 154 before starting a scan. Then, follow these steps to begin a scan.

Step	Action
1	Access the !DB/SMU Primary Menu.
2	Type **1** in the Option field.
3	Press Enter. **Result**: !DB/SMU displays the Data Space Selection panel.
4	Complete the Data Space Selection panel by specifying the criteria for the scan. (Refer to "Using the Data Space Selection Panel to Create a Selection List" on page 81 for details.)
5	Press Enter. **Result**: !DB/SMU displays the Full DB2 Space Scans panel.
6	To request a new scan, take one of these actions on the Full DB2 Space Scans panel: • Type **N** in the Sel field next to a table space or index space, or • Type **S** in the Sel field next to a table space or index space and press Enter. Then type **S** in the "Select here ==> _ to perform a new full space scan" field of the Report Select List panel.
7	Press Enter. **Result**: !DB/SMU begins the scan.

Verifying a scan

If the dataset you selected is very large, !DB/SMU displays the Space Scan Verify panel. This panel informs you that the space is large and provides an estimate of the time the scan will take.

Refer to the chart for information on using the Space Scan Verify panel.

Task You Want to Perform	Action to Take
Continue the scan.	Press Enter.
Cancel the scan. (For large spaces, you probably want to continue the scan in batch mode.)	Type **END** on the command line or press the appropriate function key.

Monitoring the progress of a scan

If the dataset you selected takes more than a few seconds to read, !DB/SMU displays the Scan in Progress panel.

For table spaces, !DB/SMU updates the Scan in Progress panel every 2 seconds until the scan is complete. Each update of this panel shows:

- The name of the database
- The name of the table space
- The number of pages in the space
- The elapsed time of the scan
- The number of pages scanned so far
- A graphic representation of the percentage of the pages scanned so far

For index spaces, !DB/SMU makes two passes through the dataset. !DB/SMU updates the Scan in Progress panel and also displays the pass number.

Completing a scan when no errors are found

As discussed earlier in this unit, you can initiate a new scan using **N** on the Full DB2 Space Scans panel or using **S** on the Report Select List panel.

If you used **N**, !DB/SMU completes the scan and returns you to the Full DB2 Space Scans panel.

However, if you initiated a scan using **S**, and if the scan completes without finding any integrity errors, !DB/SMU displays the Space Scan Complete panel. This panel contains summary information about the table space or index space that was scanned.

This chart provides the actions to take on the Space Scan Complete panel when no errors have been found.

Task You want to Perform	Action to Take
For table spaces, display the list of reports on the Table Space Stats panel For index spaces, display the list of reports on the Index Statistics panel	Press Enter.
Return to the selection list on the Full DB2 Space Scans panel.	Type **END** on the command line or press the appropriate function key.

Completing a scan when errors are found

When !DB/SMU finds data integrity errors in the space, it displays the Space Scan Complete panel. This panel contains summary information about the table space or index space that was scanned and lists the number of error pages that was found.

This chart provides the actions to take on the Space Scan Complete panel when errors have been found.

Task You want to Perform	Action to Take
Display the errors. (To diagnose and correct the errors, see “Fixing Error Pages” on page 319.)	Press Enter.
For table spaces, display the list of reports on the Table Space Stats panel For index spaces, display the list of reports on the Index Statistics panel	Type **END** on the command line or press the appropriate function key.

Problem Resolution during a Scan

Overview

This unit describes some potential error conditions that can occur when !DB/SMU is performing an online scan and some ways in which you can correct the problems.

Authorization requirements

If your installation uses an access control facility such as RACF and you attempt to read a dataset for which you are not authorized, you will get an S913 abend. !DB/SMU intercepts the abend, but you must see your system administrator to obtain the appropriate authority if you wish to proceed further.

How !DB/SMU performs scans

DB2 does not need to be active in order for !DB/SMU to perform a scan. !DB/SMU uses VSAM control interval (CI) processing instead of DB2 to scan table spaces and index spaces. Therefore, !DB/SMU is unaware of any committed updates to a table space or index space that are in the DB2 buffers during the scan.

Possible error conditions during a scan

It is not absolutely necessary to restrict a table space or index space from DB2 processing while performing a scan. (Such restriction could take the form of STOPping the space, for example.) However, because !DB/SMU is unaware of the committed updates to a page that is in the DB2 buffer during a scan, the scan can signal integrity errors that are false alarms. An overflow row might appear to be orphaned because the page containing its pointer row was not on DASD when !DB/SMU was performing the scan.

Resolving “erroneous” errors

To determine whether an error condition !DB/SMU identifies is a false alarm, you can take one of the following actions:

- Rerun the scan at a later time and determine whether previous errors persist.
- Issue a QUIESCE WITH WRITE, which causes DB2 to write out the contents of the buffers to the data base and allows updating to continue.
- STOP the space and rerun the scan. (The space can be started for read-only access while !DB/SMU is scanning it.)

  Any errors that persist after the data base is stopped or the buffers are written to the data base are real rather than erroneous.

Chapter 10. Using the Scan Reports for Table Spaces

Introduction

This chapter contains information on using the !DB/SMU table space reports that are produced by an online scan.

Chapter Contents

Chapter Overview

Prerequisites for using this chapter

Before reviewing the information in this chapter, read the chapter "Using the Scan Function" on page 153.

Generating table space scan reports in batch

You can use batch procedures to generate reports similar to those in this chapter. For information about the batch versions, see "Using the Batch Reports for !DB/SMU" on page 591.

Organization of information in this chapter

The units in this chapter describe the table space reports generated by an online scan. Each unit describes the way in which you access the report using the !DB/SMU online interface and describes the contents of the report.

Types of scan reports for table spaces

The table space scan reports provide details about table space usage and help you evaluate the need to perform full or partial image copies or reorganize the database. This chart provides summary information about each scan report and points you to detailed information.

Table Space Report	Report Contents	Detailed Information
Pages Analysis	Summary statisitics for pages, including number of errors, updated pages, and pages needed after REORG, and the copy factor	"Pages Analysis Report" on page 169
Pointer Analysis	Severity and distrib-ution of overflow records	"Pointer Analysis Report" on page 176
Freespace Analysis	Number of pages that do and do not have free space available for rows	"Free Space Analysis Report" on page 179

Types of scan reports for table spaces (continued)

Table Space Report	Report Contents	Detailed Information
DASD Space Analysis	Summary of DASD usage for data pages, dictionary pages, overhead, free space, and unformatted pages	"DASD Space Analysis Report" on page 186
Table Rows Summary	Number of rows in each table and the table space	"Table Rows Summary Report" on page 189
Table Space Usage	Space usage for each table in the table space	"Table Space Usage Report" on page 192
Segment Statistics	Number of pages and the average distance between pages for each table in the table space	"Segment Statistics Report" on page 195
Segment Map	Distribution of the segments for all the tables in the table space	"Segment Map Report" on page 197
Dictionary Report	Statistics for the DB2 dictionary pages	"Dictionary Report" on page 201
CPR Rows Summary	Number of compressed rows in each table and the pages they occupy	"Compressed Rows Report" on page 203
Compressed Space Usage	Statistics for ESA compression and the number of pages saved	"Compressed Space Usage Report" on page 205

Using Scan Reports for Table Spaces

Overview

Each time you run the !DB/SMU scan function, it creates a series of table space reports. This unit describes how to access the reports and navigate through them.

Selecting the appropriate generation of reports

When you access the Report Select List panel, you are presented with a list of all available reports for a table space, sorted in date order with the most recent generation of reports at the top of the list. You can view these most recent reports or earlier ones.

Accessing the scan reports for table spaces

Follow these steps to access the scan reports for table spaces.

Step	Action
1	Access the Report Select List panel as described in “Viewing a Report” on page 91.
2	In the list of available reports, type **S** in the SEL column next to the reports you want to see.
3	Press Enter. **Result:** !DB/SMU displays the Table Space Stats panel.
4	On the Table Space Stats panel, type the number of the report you want to see on the command line or press the appropriate function key. If you want to view the reports in order, press Enter.

Elements of the Table Space Stats panel

This is the Table Space Stats panel.

```
TABLE SPACE STATS --------------   DB/SMU  ---  DB2=D31A  --------------------
Cmd ===> 1

                  TABLESPACE KTSS32DS ON mm/dd/yy AT 06:25 2

    SELECT ENTRY BY NUMBER, OR
     PRESS ENTER TO VIEW IN SEQUENCE


               1. PAGES        PAGES ANALYSIS REPORT
               2. PTRS         OVERFLOW RECORDS POINTERS
               3. FREESPACE    ANALYSIS OF PAGES FREESPACE
               4. DASD         SUMMARY OF DASD SPACE USAGE     3
               5. ROWS         TABLE ROWS SUMMARY
               6. ROWSPACE     ANALYSIS OF TABLE ROWS SPACE
               7. SEG STATS    SEGMENT STATISTICS              4
               8. SEG MAP      SEGMENT MAP
               9. DICT         DICTIONARY REPORTS
               10.CPR SUMM     CPR ROWS SUMMARY                5
               11.CPR USAGE    COMPRESSED SPACE USAGE

No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
 1 Pages         2 Pointers            3 Freespace              4 DASD space  6
 5 Rows          6 Rows space          7 Segment Stats          8 Segment Map
 Optional CPR Reports:   9 Dict       10 CPR Rows    11 CPR Usage
```

1 Command line; type the number of the report you want to view here.

2 Name of table space for which the reports have been generated, and the date and time of this generation of reports

3 Basic set of available scan reports for table spaces

4 Additional reports available for segmented table spaces. These reports are not listed on the Table Space Stats panel unless you are working with a segmented table space.

5 Additional reports available for compressed table spaces. These reports are not listed on the Table Space Stats panel unless you are working with a compressed table space.

6 Commands and PF keys that are active on this panel. The actions of these keys are described in the next chart.

Actions you can take on scan report panels

This chart describes the actions you can take on any scan report panel to move among the various available scan reports.

Action	Result
Press Enter.	!DB/SMU moves through the available reports page by page in the order in which they are listed on the screen.
Type the option number for a report on the command line and press Enter.	!DB/SMU directly displays the selected report.
Type **N** on the command line	!DB/SMU displays the next report on the list and bypasses remaining pages of the current report.
Type **P** on the command line	!DB/SMU displays the previous report on the list and bypasses remaining pages of the current report.
Type **T** on the command line	!DB/SMU returns you to the Table Space Stats panel, the menu panel (or top panel in the hierarchy) for the scan reports.

Pages Analysis Report

Overview

This unit provides information on the pages analysis report.

Purpose of the pages analysis report

This report summarizes the status of all the pages in the table space. Use the pages analysis report to do the following:

- Determine how many pages in the table space contain errors
- Obtain page use statistics for all the pages in the table space
- Determine how many pages running the REORG utility can save
- Obtain an adjustment value for efficient DASD use
- Determine the amount of activity since the last image copy
- Evaluate the need for performing an image copy

Accessing the pages analysis report

To access the pages analysis report, select option 1 on the Table Space Stats panel.

Elements of the pages analysis report

```
TABLE SPACE STATS --------------   DB/SMU  ---  DB2=D31A  ---------------------
Cmd ===>
                  TABLESPACE KTS02SCG ON mm/dd/yy AT 13:42
                                                                        1
1. PAGES ANALYSIS                                                 4  PAGES WITH ERRORS
                                  NUMBER        PCT OF
                                OF PAGES       ALL PGS
                                ---------      -------
2          PAGES WITH ROWS          93         11.74%
3        DB2 CONTROL PAGES           2          0.25%
4     DB2 DICTIONARY PAGES          16          2.02%
5              EMPTY PAGES           6          0.76%
6        UNFORMATTED PAGES         681         85.98%

                     TOTAL         792        100.00%
7    ESTIMATED AFTER REORG         111

8            UPDATED PAGES           2          0.25%           11
9           UPDATED TRACKS           1                   COPY FACTOR  =   1.5
10     UPDATED EMPTY PAGES           0
                                              PRESS ENTER FOR UPDATE GRAPH 12
No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
 1 Pages         2 Pointers          3 Freespace             4 DASD space
 5 Rows          6 Rows space        7 Segment Stats         8 Segment Map
 Optional CPR Reports:   9 Dict       10 CPR Rows    11 CPR Usage
```

1 Number of pages with possible errors. (No message appears if the space scan found no errors)

2 Number of pages containing data

3 Number of DB2 control pages (DB2 header page and space map pages)

4 Number of ESA compressed space dictionary pages (displayed only if dictionary pages are present)

5 Number of empty formatted pages that no longer hold data or that never held data

6 Number of empty, unformatted pages never used by DB2

7 Number of pages estimated for the table space if it is reorganized immediately after the scan (based on current free space and free page values at the time of scan). Variable length rows may skew these results.

8 Number of pages that would be copied for an incremental image copy

9 Number of updated tracks

10 Number of pages whose rows have been deleted since the last image copy

Elements of the pages analysis report (continued)

11 Percentage of updated tracks (indicates whether or not an image copy is needed)

12 Indicator for update graph (Displayed only when there is sufficient data to build the graph.) Press Enter to see the graph.

Interpreting the pages analysis report to optimize DASD use

Review the number of pages you have currently allocated to the space. You can find this value in the Total field on this panel. (In the example panel, the value in the Total field is 792.) To use DASD more efficiently:

- Determine whether this is a heavily updated table by examining the copy factor. You can find detailed information about the copy factor in the unit "Copy Factor" on page 100.
  - If the table is not heavily updated (that is, it has a low copy factor), adjust the DASD allocated to the table space to match the value in the Estimated After Reorg (**7**) field. Multiply this value by 4 since the DASD is allocated in kilobytes.
  - If the table is heavily updated, you may want to allocate more space than indicated by the Estimated After Reorg field to provide sufficient room to include changes while keeping overflow records to a minimum.
  - If the table space contains variable length rows, the value in the Estimated After Reorg field may need to be adjusted based on your experience.

  DB2 rounds the value you obtain to the nearest track or cylinder boundary.
- Adjust the PRIQTY or primary allocation value as discussed below.

Determining the value to adjust

Use this table to determine what value you want to adjust based on your previous analysis.

Type of Table Space	Action
For a table space in a storage group	Adjust the PRIQTY value in the table space definition.
For all other table spaces	Adjust the primary allocation value.

Evaluating the need for an image copy

Use the pages analysis report to evaluate the need for making an image copy. Review the statistics to determine whether an image copy is required and whether you should perform a full or incremental copy.

- A high copy factor indicates that you should consider performing an image copy of the dataset.
- The number of updated tracks determines whether you should consider running an incremental image copy as opposed to a full image copy.

  If the number of updated tracks is high, the amount of time you could save by performing an incremental image copy rather than a full image copy is not significant.

Graph for Updated Pages

Overview

The graph for updated pages is produced only when there is sufficient data on the pages analysis report to produce the graph. This unit describes the graph.

Purpose of the graph for updated pages

The graph for updated pages summarizes the state of the entire DB2 table space into a single panel and shows the distribution of updated pages in the dataset. Use this graph to do the following:

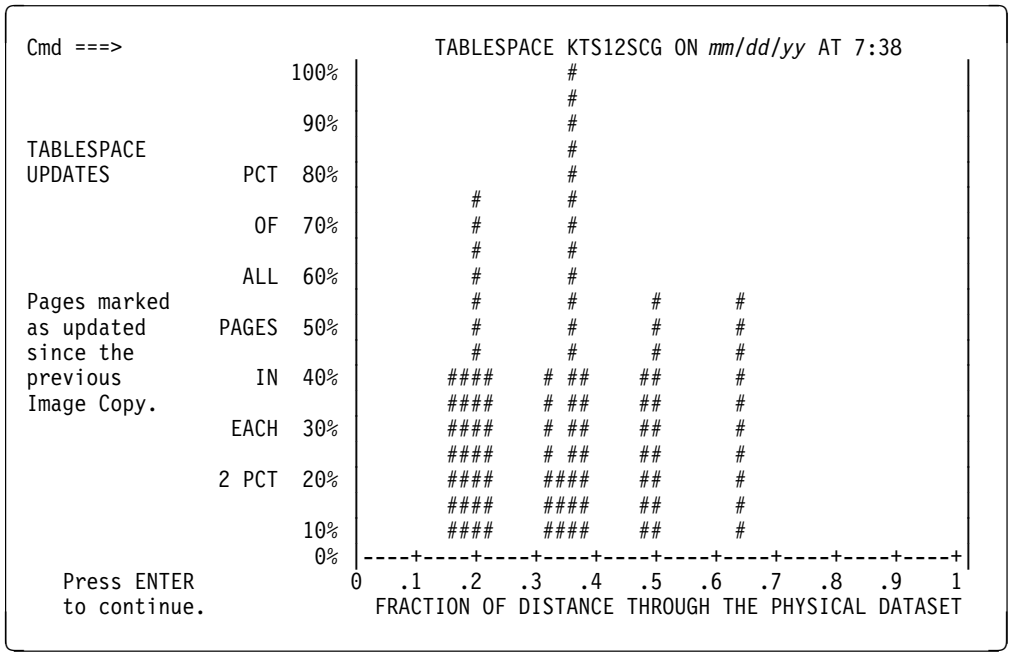
- View the distribution of updated pages in the dataset
- Look for patterns that indicate problems with the design or use of a table space.
- Compare the general pattern of update activity with the free space graph in the unit "Example usable free space graph" on page 182 to see whether the free space is adequate for the updates being performed.

Accessing the graph for updated pages

If a graph for updated pages is available, a message to that effect is displayed just above the PF keys on the Pages Analysis panel. Press Enter to display the graph.

Elements of the graph for updated pages

This figure contains a sample graph that shows the distribution of updated pages.



The graph axes provide the following information:

- Each of the grid marks at the bottom, labeled .1, .2, and so forth, represents 10% of the physical dataset. Thus, each dash (-) represents 2% of the physical dataset.
- Each column shown on the graph represents the percentage of updated pages in a specific 2% of the dataset. For a 100-cylinder dataset, each column represents 2 cylinders.

Interpreting the graph for updated pages

The pattern of updates that the graph illustrates often reveals problems with the design or use of a table space. It is a good idea to compare the general pattern of update activity as shown in this graph with the free space graph to determine whether the free space is adequate for the number of updates being performed.

Interpreting the example graph

The example graph shows the following:

- More than 50% of the update activity occurred on 18% of the dataset. (This activity is represented by the 9 columns of # symbols reaching the 50% level or more on the graph.)
- Most of the updates occur in the first half of the physical dataset.

A graph with this pattern shows that there are four relatively narrow hot spots in this data space. Such a graph might occur when, for example, a bank assigns special account numbers with the same or a similar account prefix to their Christmas Club accounts. These special accounts would generate a great deal of activity during a relatively small portion of the calendar year resulting in hot spots in the data space.

Pointer Analysis Report

Overview

This unit provides information on the pointer analysis report.

Purpose of the pointer analysis report

The pointer analysis report summarizes the number of pointer records and overflow records. Use this report to do the following:

- Determine how many pointer records exist for the table space
- Determine how many overflow records exist for the table space
- Evaluate data integrity
- Evaluate the need for reorganization based on pointer and overflow record information

Background about the creation of pointer records

The following sequence of events occurs when DB2 creates a pointer record:

1. A table row is updated.
2. The updated row is longer than the original row. There is insufficient free space in the original page for the new, longer row.
3. DB2 writes the updated row to a new page as an overflow record and replaces the original row with a 6-byte pointer record.

To place the row into the appropriate location in the table, you must either reload the table space or run the DB2 REORG utility.

Accessing the pointer analysis report

To access the pointer analysis report, select option 2 on the Table Space Stats panel.

Elements of the pointer analysis report

```
TABLE SPACE STATS --------------   DB/SMU  ---  DB2=D31A  -------------
Cmd ===>
                  TABLESPACE KTS02SCG ON mm/dd/yy AT 3:42

2. POINTER ANALYSIS

 1        POINTER RECORDS               0
 2       OVERFLOW RECORDS               0
 3   PTR TO REC NEXT PAGE               0
 4   PTR TO REC SAME TRK                0
 5   AVG PAGES  PTR-->REC               0


No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
 1 Pages         2 Pointers            3 Freespace              4 DASD space
 5 Rows          6 Rows space          7 Segment Stats          8 Segment Map
 Optional CPR Reports:   9 Dict       10 CPR Rows    11 CPR Usage
```

1 Number of pointer records in the table space. (Should equal the number of overflow records or a possible error is indicated)

2 Number of overflow records in the table space. (Should equal the number of pointer records or a possible error is indicated)

3 Number of times an overflow record is in the page next to the page that contains its pointer record

4 Number of times an overflow record is not in the page next to the page that contains its pointer record but is in another page on the same track

5 Average distance in pages between the pointer records and their overflow records

Error conditions

!DB/SMU checks for an exact match between each pointer record and its overflow row and issues error messages for any nonmatching record. If the value in the Pointer Records field does not equal the value in the Overflow Records field, it is possible that integrity errors exist in the data base. It is also possible that some errors !DB/SMU reports are false because DB2 was accessing the table space at the time of the scan. For information about this condition, see “Problem Resolution during a Scan” on page 160 in the chapter “Using the Scan Function.”

Interpreting the performance impact of overflow and pointer records

Consider the following factors when assessing the impact of overflow and pointer records and the need to reorganize a table space:

- Accessing each of these records requires an extra page read. If the number of pointer records is very large, performance suffers.
- The probability and the cost of extra I/Os going from the pointer record to the overflow record is related to the distance between these records.

  !DB/SMU analyzes the distance between each pointer record and its corresponding overflow row and computes the average distance in pages between each pair. If the overflow and pointer records occur on the same track, the effect is minimized and there is not as strong a need for reorganization.
- The use of the table space determines the effect of these records. If the table space is sequentially scanned, pointer records and overflow rows can significantly slow the scan.

Free Space Analysis Report

Overview

This unit provides information about the free space analysis report.

Purpose of the free space analysis report

The free space analysis report contains an analysis of pages with free space. Use this report to do the following:

- Obtain information on the quantity of pages that have no free space
- Obtain information on the quantity of pages that have free space for a
  - Short row
  - Average row
  - Long row

  !DB/SMU computes the value for the short, average, and long row sizes as discussed in "Reorg Factor" on page 98.
- Evaluate the likelihood that DB2 will find room to insert a row in a page of the table space.

Accessing the free space analysis report

To access the free space analysis report, select option 3 on the Table Space Stats panel.

Elements of the free space analysis report

```
TABLE SPACE STATS --------------  DB/SMU  ---  DB2=D31A  -------------
Cmd ===>
                  TABLESPACE KTS02SCG ON mm/dd/yy AT 13:42

3. FREESPACE ANALYSIS

                              NUMBER        PCT OF
                            OF PAGES       ALL PGS        SIZE
                           ---------       -------       -----
     PAGES WITH:
1             NO ROOM            34         4.29%       4,096

2  ROOM FOR SHORT ROW            42         5.30%          23

3 ROOM FOR AVERAGE ROW           88        11.11%          48

4    ROOM FOR LONG ROW          628        79.30%          74

5 FREE SPACE SPECIFIED  1%                          8 REORG FACTOR =   4.3
6 FREEPAGE SPECIFIED    0                    PRESS ENTER FOR FREESPACE GRAPH 9
7 ESA COMPRESSED: YES
No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
   1 Pages   2 Pointers    3 Freespace   4 DASD space   5 Rows   6 Rowsspace
   Optional CPR Reports:   9 Dict       10 CPR Rows    11 CPR Usage dd
```

If the value for any of items **1** through **4** is 0, then there is only 1 table in the table space.

1 Number of pages without free space

2 Number of pages that have room only for a row of the shortest row size of all the tables in the table space

3 Number of pages that have room for a row of the average row size of all the tables in the table space

4 Number of pages that have room for a row of the longest row size of all the tables in the table space

5 Free space value that was in the DB2 catalog at the time the extract was run. (If this value is zero, the free space graph is generated on the current specifications.)

6 Free page value that was in the DB2 catalog at the time the extract was run. (If this value is zero, the free space graph is generated on the current specifications.)

7 Status of ESA compression at the time the extract was run (displayed only if the space is compressed)

8 Percentage of pages with no room (The reorg factor is an exact indicator of the fullness of the table space and its need to be reorganized.)

Elements of the free space analysis report (continued)

9 Indicator for free space graphs—graph of usable free space and graph of total free space (displayed only when there is sufficient data to build the graphs) Press Enter to see the graphs.

Source of additional information

Detailed information about the reorg factor and how !DB/SMU computes the values associated with row length can be found in “Reorg Factor” on page 98.

Interpreting the free space analysis report

Examine the reorg factor and the lines that indicate how much room is available for each type of row. Review the number of pages with no room.

The reorg factor is a weighted average of the percentages of each type of page and indicates the need for reorganization. The following chart explains its meaning.

Reorg Factor Size	Meaning
Small	The table space has a great deal of free space.
Large	The table space cannot accept many inserts without adding more extents.

Graph of Usable Free Space

Overview

This unit describes the usable free space graph and suggests ways it can be interpreted.

Purpose of the usable free space graph

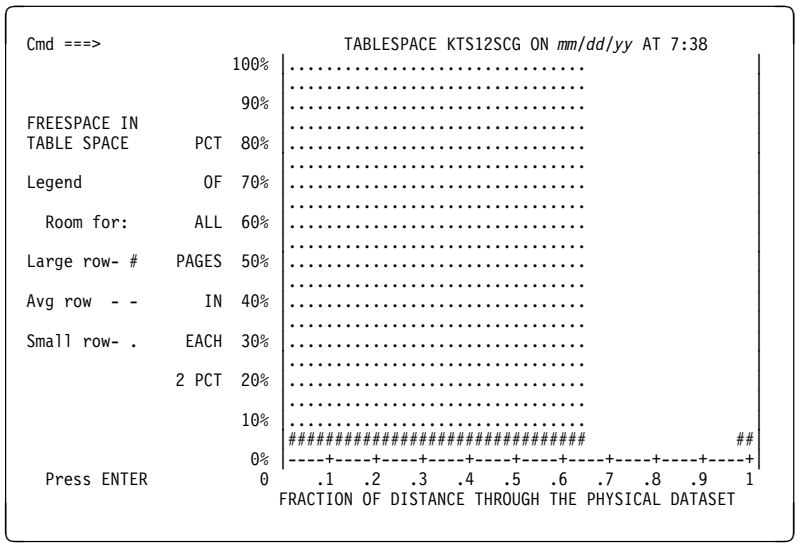
This graph shows the relative distribution of *usable* free pages in the DB2 table space.

Accessing the usable free space graph

To view the graph of usable free space, press Enter on the free space analysis report. *This graph is shown only if the table space is large enough to build a meaningful graph.*

Example usable free space graph

The is an example usable free space graph. !DB/SMU divides the table space into equal sections and computes the average free space in each section.



Reading the usable free space graph

For good update performance, free space should be evenly distributed across the dataset so that rows can be added evenly. The graph elements provide the following information:

- The values shown represent the amount of room that DB2 recognizes and can use to hold new rows, not the actual free space in the table.
- Each of the grid marks at the bottom, labeled .1, .2, and so forth, represents 10% of the physical dataset.
- Each column of the graph represents 2% of the dataset.
- Peaks in the graph show areas where there is abundant free space.
- Valleys in the graph show where there is a shortage of free space.

Interpreting the graph

The usable free space graph compresses the entire table space to 50 columns, each column representing the composite of 2% of the dataset's space. It shows, in a single picture, the distribution of free space pages in the dataset.

If the table space is heavily updated, which you can determine by reviewing the REORG factor on the free space analysis report, and especially if there is a clustering index on the table, the free space should be evenly distributed across the dataset to achieve optimum performance.

Interpreting the example graph

The example free space graph illustrates these points:

- The first 65% of the example graph has room primarily for short rows.
- Less than 10% of that space will hold large rows.
- The remaining 35% of the example graph has room for average rows.

A graph like this one may have a negative impact on performance if the data in the first 65% is heavily updated, resulting in a significant number of overflow records.

Usefulness of the graph for segmented table spaces

If your table space is segmented and contains multiple tables, the usefulness of this graph is limited as it does not show to which table the pages belong.

Graph of Total Free Space

Overview

This unit describes the total free space graph and suggests ways it can be interpreted.

Purpose of the total free space graph

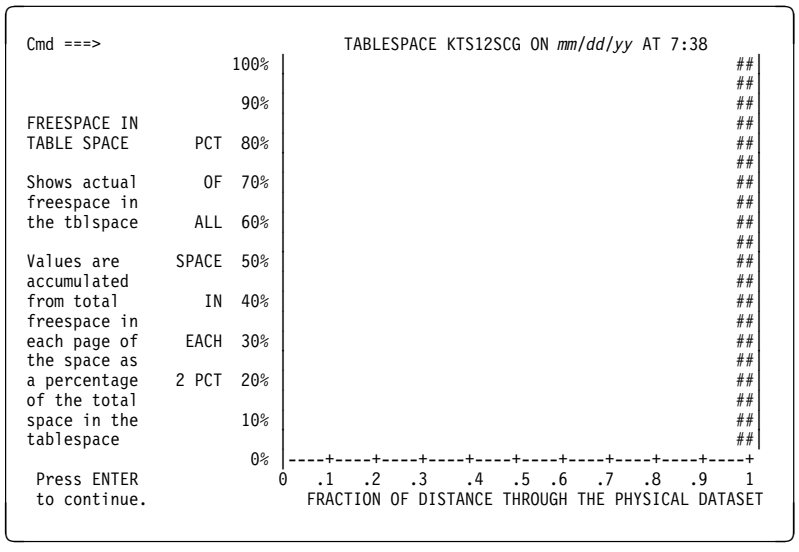
This graph shows the relative distribution of *actual* free space pages in the DB2 table space.

Accessing the total free space graph

Press Enter on the usable free space graph to display the total free space graph. *This graph is shown only if the table space is large enough to build a meaningful graph.*

Example total free space graph

This is an example graph of all of the free space in a given table space. !DB/SMU divides the table space into equal sections and computes the average free space in each section.



Reading the total free space graph

For good update performance, free space should be evenly distributed across the dataset so that rows can be added evenly. The graph elements provide the following information:

- The values shown represent the actual amount of free space in the pages. DB2, however, may be unable to use all this space to hold new rows.
- Each of the grid marks at the bottom, labeled .1, .2, and so forth, represents 10% of the physical dataset.
- Each column of the graph represents 2% of the dataset.
- Peaks in the graph show areas where there is abundant free space.
- Valleys show where there is a shortage of free space.

Interpreting the graph of total free space

In the example graph, the table space is full with the exception of the last 4% of the cylinders. There are several possible conclusions you can draw from this graph:

- If the table space experiences few or no updates, then this pattern indicates that DASD space may be over-allocated. You can determine the pattern of updates for this table space by referring to the copy factor for this table, which you can find on the pages analysis report.
- If the table space experiences many updates, as can be determined from the copy factor, you then want to determine whether the free space is spread evenly over the data space. In this example, the free space is found only in the last 4% of the space. You should examine a space with this type of pattern to determine whether reorganization or reallocation is indicated.

DASD Space Analysis Report

Overview

This unit provides information about the DASD space analysis report.

Purpose of the DASD space analysis report

This report classifies the DASD space occupied by the table space into separate categories. Use the DASD space analysis report to do the following:

- Determine how DASD space is being used
- Determine the amount of wasted space in the dataset
- Determine how the pages in the table space are being used

Accessing the DASD space analysis report

To access the DASD space analysis report, select option 4 on the Table Space Stats panel.

Elements of the DASD space analysis report

```
TABLE SPACE STATS -------------   DB/SMU  ---  DB2=D31A  --------------
Cmd ===>
                  TABLESPACE KSC01DB  ON mm/dd/yy AT 13:42

4. DASD SPACE ANALYSIS
                                 1           2           3           4
                              NUMBER      AMOUNT      % USED      % TOTAL
                            OF PAGES   OF SPACE       SPACE       SPACE
                           ---------   -------     -------     -------

5              ROWS DATA          178       266K      26.50%       8.40%
6   DB2 DICTIONARY PAGES           24        64K       6.37%       2.02%
7      WASTED DICTIONARY            8        32K       3.19%       1.01%
8   PAGES WITH  OVERHEAD          222        91K       9.06%       2.87%
9   PAGES W WASTED SPACE           42       167K       0.05%       0.02%
10 PAGES WITH FREESPACE           225       550K      54.82%      17.37%
11       FORMATTED EMPTY           42       167K                   5.27%
12     UNFORMATTED EMPTY            5        20K                   0.63%
13 ALLOCATED / NOT USED           541     2,164K     215.54%      68.31%

                                              14          15
       TOTAL DATASET SPACE                 3,168K        PCT FREESPACE = 85.7

No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
   1 Pages   2 Pointers    3 Freespace   4 DASD space   5 Rows   6 Rowsspace
   Optional CPR Reports:   9 Dict       10 CPR Rows    11 CPR Usage
```

1 Number of pages in this category

2 Total amount of space this category occupies

3 Percentage of the used space this catagory occupies

4 Percentage of the total space this catagory occupies

5 Total number of pages having rows with data, including pages with dropped rows, and excluding header, space map, and dictionary pages

6 Total number of pages that contain the dictionary (This line appears only for compressed table spaces.)

7 Number of wasted dictionary pages—the number of pages in the segments allocated to the dictionary that do not contain dictionary pages (This line appears only for compressed and segmented table spaces.)

8 Total number of pages in the table space that contain overhead.

9 Total number of pages in the table space that contain wasted space. A page is considered to have wasted space when the amount of free space in the page exceeds the average row size multiplied by the number of open rows.

10 Total number of pages in the table space that contain free space

Elements of the DASD space analysis report (continued)

11 Total number of pages in the table space that are formatted but empty. These pages are present when a page once held data but all of the rows have been deleted, or as the result of a FREEPAGE specification during table space creation.

12 Total number of pages in the table space that are allocated to a table but are unformatted

13 Total number of pages in the table space that are allocated to the table space but not yet processed by DB2. This value is *not* included in the pages with free space value.

14 Size of the dataset in K-bytes

15 Percentage of free space in both unused and unformatted pages

Interpreting the DASD space analysis report

This report shows you, on a single panel, the DASD usage for a given data space. These are some suggestions for reviewing the report:

- The Wasted Dictionary Pages value applies to compressed and segmented table spaces. The number of pages shown are considered by !DB/SMU to be allocated to the dictionary. They cannot be used by any table in the table space. If you want to reclaim this space, you must change the segment size to a multiple of the total number of dictionary pages.
- Pages with Wasted Space are those having an amount of free space that exceeds the value represented by the average row size multiplied by the number of open rows. You may want to reclaim this space by reorganizing the data space.
- You can compare the % of Used Space to the % of Total Space. If the % of Used Space is small, it can indicate that DASD is over-allocated to the table space.
- If the Allocated / Not Used value is high in comparison to the balance of the values, the table space may be over-allocated and should be redefined.
- When determining whether or not a table space is over-allocated, you can also review the pages analysis report to determine the copy factor for the table space. The copy factor (percentage of updated pages) will help you determine the activity of this table space. If this is an active table space, it may require more free space.

Table Rows Summary Report

Overview

This unit provides information on the table rows summary report.

Purpose of the table rows summary report

This report summarizes information about the tables in the table space. Use the table rows summary to do the following:

- Obtain summary statistics for each table in the table space
- Obtain summary statistics for the entire table space
- Evaluate the potential benefits of eliminating dropped rows
- Evaluate the distribution of compressed rows across a table space

Accessing the table rows summary report

To access the table rows summary report, select option 5 on the Table Space Stats panel.

Elements of the table rows summary report

```
TABLE SPACE STATS --------------   DB/SMU  ---  DB2=D31A  --------------
Cmd ===>
                  TABLESPACE KTS02SCG ON mm/dd/yy AT 13:42

5. TABLE ROWS SUMMARY          1        2          3    4      5          6
                               ROWS IN PCT OF    PAGES  ROWS OVERFLOW COMPRESS
CREATOR.TABLE                    TABLE  TOTAL OCCUPIED /PAGE     ROWS     ROWS
-------------------------- ----------- ------ -------- ----- -------- --------

TOOLSTX.KTS02C22                 3,388   33.9%       15 225.9        0    3,388
TOOLSTX.KTS02C33                 3,372   33.7%       63  53.5        0        0
TOOLSTX.KTS02C44                 3,388   33.9%       15 225.9        0    3,388

 DROPPED TBLS ROWS 7                 0

 * TOTALS ** 8                   9,999  100.0%       93 107.5        0    6,666



No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
   1 Pages   2 Pointers    3 Freespace   4 DASD space   5 Rows   6 Rowsspace
   Optional CPR Reports:   9 Dict       10 CPR Rows    11 CPR Usage
```

1 Number of rows in each table

2 Percentage of the total number of rows in the table space (Any value 50% or greater is highlighted)

3 Number of individual pages in each table space that contain rows of this table

4 Number of rows in each table divided by the number of occupied pages (yields the average number of rows per occupied page)

5 Number of overflow rows belonging to each table

6 Number of compressed rows belonging to each table

7 Total number of rows from dropped tables in the table space. (Dropped tables' rows can continue to occupy space until you reorganize the table space.)

8 Summary totals for the entire table space

Interpreting the Table Rows Summary Report

This report provides an overview of the rows that comprise the data space. You can observe the following:

- The maximum number of rows per page is 255. A high number (a number approaching 200 for example) indicates that the rows in this table are probably small or medium length rows.
- If you are at or near the maximum number of rows per page (255) and you have a significant amount of space remaining on the page, this space remaining is wasted since few or no additional rows can be added to the page. In this case, you can increase your row length without increasing the amount of DASD required to hold the table.
- The totals row represents the sum of the values in each column except the ROWS/PAGE column. This column is computed by dividing the number of rows in the table by the pages occupied.

Table Space Usage Report

Overview

This unit provides information on the table space usage report.

Purpose of the table space usage report

This report summarizes detailed information about the space usage by the tables in the table space. Use the table space usage report to obtain summary statistics about the following:

- DASD usage for each table in the table space
- Row size for each table in the table space
- Pages with room for each table in the table space
- DASD usage and row size for the entire table space

Accessing the table space usage report

To access the table space usage report, select option 6 on the Table Space Stats panel.

Elements of the table space usage report

```
TABLE SPACE STATS --------------   DB/SMU  ---  DB2=D31A  --------------
Cmd ===>
                  TABLESPACE KTS02SCG ON mm/dd/yy AT 13:42

6. TABLES SPACE USAGE               1         2        3       4        5        6
                             ROWS IN      ROWS  PCT OF  AVG LEN    COMP   PAGES
CREATOR.TABLE                  TABLE     SPACE   TOTAL  PER ROW   RW SZ  W ROOM
-------------------------   -------- --------  ------  -------   -----  -------

TOOLSTX.KTS02C22               3,388       26K    9.8%      7.9      66      678
TOOLSTX.KTS02C33               3,372      217K   81.7%     66.0      66      678
TOOLSTX.KTS02C44               3,388       26K    9.8%      7.9      66      678

 DROPPED ROWS 7                    0

 * TOTALS **           8       9,999     266K  100.0%     27.2



No. Report No.    N Next Report    P Previous Report    T Top    PF15 END
   1 Pages   2 Pointers    3 Freespace   4 DASD space   5 Rows   6 Rowsspace
   Optional CPR Reports:   9 Dict       10 CPR Rows    11 CPR Usage
```

1 Number of rows in each table

2 Amount of DASD space, in K-bytes, used by this table

3 Percentage of DASD space used by each table (Any value 50% or greater is highlighted.)

4 Average size of rows in each table computed by dividing the accumulated length of all rows in the table by the number of rows. Use this value for space estimations.

5 Sum of the defined lengths of all the columns in each table (VARCHAR columns are counted full size.)

6 Number of pages in the table space that have room for the computed row size

7 Total number of rows from dropped tables in the table space. (Dropped tables' rows can continue to occupy space until you reorganize the table space)

8 Summary totals for the entire table space

Interpreting the Table Space Usage Report

In reviewing this display, you can determine the following:

- The Pct of Total column indicates what percentage of the total data space a particular table occupies.
- Comparing Avg Len per Row with Comp Row Size indicates how well rows for a given table are being compressed. In this example, the computed row size for table KTS02C22 is 66 bytes before compression. After compression, the row size is 7.9 bytes. In contrast, table KTS02C33 has no compressed rows; both columns have a value of 66 bytes.
- The Pages with Room column does not show empty pages. Rather, it indicates the number of pages that can accommodate at least one more non-compressed row of the computed row size for the table.

Segment Statistics Report

Overview

This unit provides information about the segment statistics report.

Purpose of the segment statistics report

This report provides summary segment statistics. *It is provided only for segmented table spaces.* Use the segment statistlics report to do the following:

- Determine table size, compactness, and order of creation
- Identify tables that are candidates for reorganization

Accessing the segment statistics report

To access the segment statistics report, select option 7 on the Table Space Stats panel.

Elements of the segment statistics report

```
TABLE SPACE STATS --------------   DB/SMU  ---  DB2=D31A  --------------
Cmd ===>
                  TABLESPACE KTS02SCG ON mm/dd/yy AT 13:42

7. SEGMENT STATISTICS                        1       2        3
                                                              MEAN PAGE
            AUTHID   TABLE NAME          OBID    PAGES   SEPARATION
            -------- ------------------  ----  -------  ----------
            TOOLSTX  EMP                   10       14        7.15
            TOOLSTX  PROJ                  11       62        1.07
            TOOLSTX  STAFF                 12       14        5.62
```

1 Object identifier for the table

2 Number of pages containing rows of the table

3 Average physical distance between pages containing rows of the table

Interpreting the Segment Statistics Report

This chart contains information on interpreting the statistics provided by the segment statistics report.

To Evaluate...	Use the Statistics for...	Explanation
Creation order	OBID (Assists in interpreting the segment map report.)	DB2 allocates OBIDs sequentially when the table is created. Generally, the lower the value, the older the table. DB2 does, however, reuse OBIDs. **Example**: Table EMP with OBID 10 was created first; table STAFF with OBID 12 was created last.
Size	Pages	The lower the value, the smaller the table. **Example**: Table EMP with 10 pages is the smallest; table PROJ with 62 pages is the largest.
Compactness	Mean page separation	The lower the value, the more compact the table. **Example**: Table PROJ is the most compact. An average page separation of 1.07 shows that its 62 pages are nearly physically contiguous, without any intervening segments from other tables.
Separation	Mean page separation	The higher the value, the more scattered the table. **Example**: Tables EMP and STAFF are scattered because their pages that contain data rows are separated by an average of 7.15 pages and 5.62 pages respectively.
Candidates for reorganization	Pages and mean page separation	The largest and most scattered tables require the most processing time.

Segment Map Report

Overview

This unit provides information on the segment map report.

Purpose of the segment map report

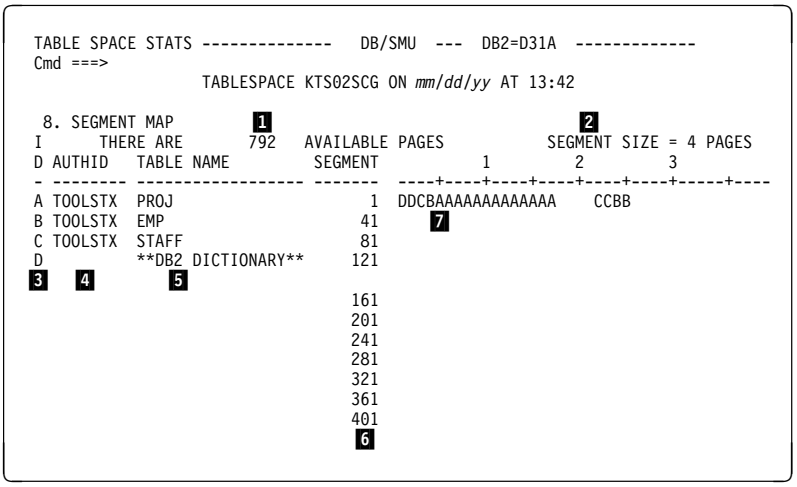
This report maps the segment locations in the table space. *It is only provided for segmented table spaces if there is enough data on updates to build the map.* The segment map shows how segments in the table space are allocated to different tables. Use this report to do the following:

- Locate the segments for different tables in the table space
- Locate unassigned segments in the table space
- Determine whether a table was dropped or deleted
- Determine which tables are the most and least populous in terms of pages
- Locate the DB2 compression dictionary and the segments used

Accessing the segment map report

To access the segment map report, select option 8 on the Table Space Stats panel.

Elements of the segment map report



```
TABLE SPACE STATS --------------   DB/SMU  ---  DB2=D31A  -------------
Cmd ===>
                  TABLESPACE KTS02SCG ON mm/dd/yy AT 13:42

 8. SEGMENT MAP         1                                2
I      THERE ARE       792   AVAILABLE PAGES           SEGMENT SIZE = 4 PAGES
D AUTHID   TABLE NAME         SEGMENT           1         2         3
- -------- ------------------ -------  ----+----+----+----+----+----+-----+----
A TOOLSTX  PROJ                    1  DDCBAAAAAAAAAAAAA    CCBB
B TOOLSTX  EMP                    41      7
C TOOLSTX  STAFF                  81
D          **DB2 DICTIONARY**     121
3   4        5
                                  161
                                  201
                                  241
                                  281
                                  321
                                  361
                                  401
                                   6
```

1 Pages available for defining a new table in a segmented table space (Displayed only for a segmented table space)

2 Segment size for the table space or DB2 compression dictionary expressed as a number of pages

3 Identification letter assigned to each table that is based upon the number of pages in the table

4 Authorization ID

5 Table or DB2 dictionary name

6 Number of the segment starting the map row

7 Segment map information

Interpreting the segment map report (left side)

The left half of the segment map associates an identifier (in the ID column) with an authorization ID and a table name. Use these guidelines to interpret the left-hand side of the panel.

- The table that has the largest number of segments is assigned to the ID *A*; the table with the second largest number of segments is assigned to the ID *B*, and so forth.
- If more than 26 tables exist in the table space, numeric and national characters can also be used as table IDs.
- If the number of tables exceeds available characters, a question mark is assigned to the least frequently occurring tables.

Interpreting the segment map report (right side)

The right half of the segment map actually maps segments of the tables listed on the left side to positions in the table space. Use these guidelines to interpret the right-hand side of the panel.

- Each position in the map represents a single segment in the table space.
- When a specific position in the segment map contains an ID such as *A* or *B*, it indicates that this position is the actual location of a segment from the table that this ID represents.
- Blanks in the map denote unassigned positions (that is, positions that do not contain a segment from one of the tables).

Additional information on reading the map follows.

Reading the segment map information

The following chart provides information in reading the segment map.

Map Content...	Explanation
Shows contiguous segments	Table IDs for contiguous segments occupy side-by-side locations. **Example**: The A-segments mapped in row 1 belong to table PROJ.
Indicates table size	Larger tables have more mapped segments than do smaller tables. **Example**: The character *A* represents the table PROJ. You can determine that it is the largest table, since the segment map shows 15 A-segments. Tables EMP and STAFF are smaller tables. Each of these tables has three segments mapped (as shown by three occurrences in the segment map of the characters *B* and *C* respectively.
Indicates when rows were added to a table	DB2 allocates the first segment for a table when the table is created. Additional segments are allocated as needed. **Example**: In row 1, a C-segment comes before the first A-segment. This indicates that rows were added to the table EMP (ID=C) before the table PROJ (ID=A) was created.
Indicates a mass delete of data or a dropped table	Gaps in a map show that a table was subjected to a mass delete or a DROP command. (The names of deleted tables are listed in the Segment Statistics report but not on the map; names of dropped tables are removed from both reports.) **Example**: The segment gap between the A and C segments in row 1 reveals that a delete or drop action occured. The matching lists of names for both reports indicate a dropped table.

Dictionary Report

Overview

This unit provides information on the dictionary report.

Purpose

This report provides page usage information about the DB2 dictionary. *It is only provided for compressed table spaces.* Use the dictionary report to determine the following:

- Number and size of dictionary pages
- Number of pages required to store the DB2 dictionary
- Maximum entries per dictionary page

Accessing the dictionary report

To access the dictionary report, select option 9 on the Table Space Stats panel.

Elements of the dictionary report

```
TABLE SPACE STATS -----------   ---   DB/SMU  ---  DB2=D31A  ------     ------
Cmd ===>
                  TABLESPACE KTS02SCG ON mm/dd/yy AT 13:42

 9. DICTIONARY REPORT

  TOTAL DICTIONARY PAGES: 1              16

  MAX ENTRIES PER PAGE:   2             508

  DICTIONARY PAGE SIZE:   3               4

  TOTAL 4K MEMORY PAGES REQUIRED:   4     16
```

1 Number of dictionary pages counted after the scan

2 Number of dictionary entries—508 for 4 K pages, 4,902 for 32 K pages

3 Page size—4 K or 32 K

4 Total number of pages required to store the dictionary

Interpreting the dictionary report

The example report indicates that the DB2 dictionary is written to 4K pages, and that the total amount of storage required to hold the dictionary is 64K (**4** 16 pages x 4K).

Compressed Rows Report

Overview

This unit provides information on the compressed rows report.

Purpose

This report provides usage information about compressed space usage. *It is only provided for compressed table spaces.* Use the compressed rows report to determine the following:

- Total number of rows in each table
- Quantity of rows in each table that are compressed
- Number of occupied pages in each table
- Percentage of rows in this table that are compressed
- Summary statistics for all the compressed rows in the table space

Accessing the compressed rows report

To access the compressed rows report, select option 10 on the Table Space Stats panel.

Elements of the compressed rows report

```
TABLE SPACE STATS --------------   DB/SMU  ---  DB2=D31A  -------------
Cmd ===>
                  TABLESPACE KTS02SCG ON mm/dd/yy AT 13:42

10. CPR ROWS SUMMARY              1          2          3          4
                                ROWS IN   COMPRESSED  CPR ROW    PAGES
CREATOR.TABLE                    TABLE      ROWS       PCT      OCCUPIED
--------------------------- ---------- ---------- -------- ----------

TOOLSTX.KTS02C22                 3,388      3,388   100.00%         15
TOOLSTX.KTS02C33                 3,372          0                   63
TOOLSTX.KTS02C44                 3,388      3,388   100.00%         15

 DROPPED ROWS  5

** TOTALS **   6                 9,999      6,666    66.67%         93



 No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
    1 Pages   2 Pointers    3 Freespace   4 DASD space   5 Rows   6 Rowsspace
    Optional CPR Reports:   9 Dict       10 CPR Rows    11 CPR Usage
```

1 Total number of rows in each table

2 Number of rows in each table that are compressed

3 Percentage of rows in each table that are compressed

4 Number of occupied pages in each table

5 Number of rows from dropped tables

6 Summary totals for the entire table space

Interpreting the compressed rows report

The example report indicates that ESA compression is in effect for two of the three tables in the table space. It also indicates that no rows have been dropped.

Compressed Space Usage Report

Overview

This unit provides information on the compressed space usage report.

Purpose of the compressed space usage report

This report summarizes usage statistics for compressed rows. *It is only provided for compressed table spaces.* Use the compressed space usage report to obtain the following space usage information:

- Total rows in the table space
- Dictionary pages
- Percentage of free pages
- Estimated pages used for the table space
- Pages without compression
- Percentage of pages saved by compression

Accessing the compressed space usage report

To access the compressed space usage report, select option 11 on the Table Space Stats panel.

Elements of the compressed space usage report

```
TABLE SPACE STATS --------------   DB/SMU  ---  DB2=D31A  -------------
Cmd ===>
                  TABLESPACE KTS02SCG ON mm/dd/yy AT 13:42

11. CPR USAGE REPORT

 1   ROWS IN SPACE:                          9,999

 2   DICTIONARY PAGES:                         16

 3   FREE SPACE (PCTFREE):                1%

 4   FREE PAGE (FREEPAGE):                 0

 5   PAGES WITH ESA COMPRESSION:              111

 6   EST. PAGES W/O ESA COMPRESSION:          193

 7   PERCENTAGE OF PAGES SAVED:                42 %

No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
   1 Pages   2 Pointers    3 Freespace   4 DASD space   5 Rows   6 Rowsspace
   Optional CPR Reports:   9 Dict       10 CPR Rows    11 CPR Usage
```

1 Total number of rows in the table space

2 Number of dictionary pages in the table space (These pages control compression and decompression.)

3 Percentage of pages to be left free

4 Number of pages between free pages

5 Number of pages having data rows (exclusive of dictionary pages)

6 Estimate of required pages if the table space was not compressed. The estimate is based on an average row size and could be skewed if the row contains VARCHAR data.

If the value for Percentage of Pages Saved (**7**) is negative, then this field contains the text ***NEGATIVE COMPRESSION***.

7 Percentage of pages in a table space or partition that was saved by compression. The estimate is based on an average row size and could be skewed if the value in **6** is skewed.

Interpreting the compressed space usage report

The example report shows that you have saved 42% of the pages in the table space or partition through the use of ESA compression. It also shows that the table space has only 1% free space.

Chapter 11.
Using the Scan Reports for Index Spaces

Introduction

This chapter contains information on using the index space reports that are produced by an online scan.

Chapter Contents

Chapter Overview

Prerequisites for using this chapter

Before reviewing the information in this chapter, read the chapter “Using the Scan Function” on page 153.

Organization of information in this chapter

This chapter describes each of the online index space reports generated by the scan function. It begins with an introduction to accessing these reports from the online interface and then describes each of the reports.

Online help

Online help provides additional information about the reports and their fields. Use the HELP command, or press the HELP function key to display the panel help.

About Scan Reports for Index Spaces

Overview

This unit provides information about the different online index space reports that !DB/SMU generates.

Producing index space reports in batch

You can use batch procedures to generate index space reports similar to those in this chapter. For information about the batch versions, see "Using the Batch Reports for !DB/SMU" on page 591.

Types of reports for index spaces

The scan reports for index spaces provide detailed information about index space usage and help you evaluate the need for database reorganization. Refer to the chart for summary information about these reports.

Index Space Report	Report Contents	Detailed Information
Index Keys Summary	Quantities of different types of records, keys, levels, and leaf pages in the index space	"Index Keys Summary Report" on page 214
Index Keys Distribution	Distribution of pages with room for keys, records per key, and the 10 most frequently used keys	"Reports for Index Keys Distribution" on page 218
Pages Analysis	Quantity of different types of pages in the index space	"Pages Analysis Report" on page 226
Usage Graph	Distribution of used index pages, overhead, free space, and unformatted pages	"Usage Graph" on page 229
Freespace Graph	Distribution of free space in the index	"Free Space Graph" on page 231

Types of reports for index spaces (continued)

Index Space Report	Report Contents	Detailed Information
Subpages Summary	Statistical totals for subpages in the index; applies only to type 1 indexes	"Subpages Summary Report for Type 1 Indexes" on page 233 space
Leaf Pages Summary	Statistical totals for leaf pages in the index; applies only to type 2 indexes	"Leaf Pages Summary Report for Type 2 Indexes" on page 235 space
Index Level Summary	Key factors regarding index levels	"Index Levels Summary Report" on page 238

Using Scan Reports for Index Spaces

Overview

Each time you run the !DB/SMU scan function, it creates a series of index space reports. This unit describes how to access the reports and navigate through them.

Selecting the appropriate generation of reports

When you access the Report Select List panel, !DB/SMU presents you with a list of all available reports for an index space, sorted in date order with the most recent generation of reports at the top of the list. You can view these most recent reports or earlier ones.

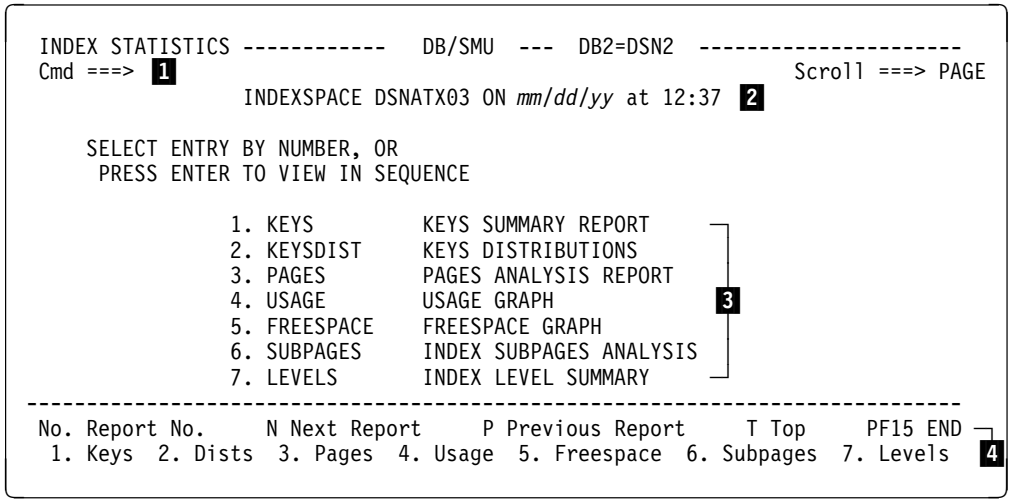
Accessing the scan reports for index spaces

Follow these steps to access the scan reports for index spaces.

Step	Action
1	Access the Report Select List panel as described in “Viewing a Report” on page 91.
2	In the list of available reports, type **S** in the SEL column next to the reports you want to see.
3	Press Enter. **Result:** !DB/SMU displays the Index Statistics panel.
4	On the Index Statistics panel, type the number of the report you want to see on the command line or press the appropriate PF key. If you want to view the reports in order, press Enter.

Elements of the Index Statistics panel

This is the Index Statistics panel.



1 Command line; type the number of the report you want to view here.

2 Name of index space for which the reports have been generated, and the date and time of this generation of reports If this were a type 2 index, this line would read:

TYPE 2 INDEX DSNATX03 ON *mm/dd/yy* AT 12:37

3 Available scan reports for index spaces. If these scan reports were for a type 2 index, option 6 would read as follows:

6. LEAF PAGES INDEX LEAF PAGE ANALYSIS

4 Commands and PF keys that are active on this panel. The actions of these keys are described in the next chart.

Actions you can take on scan report panels

This chart describes the actions you can take on any scan report panel to move among the various available scan reports.

Action	Result
Press Enter.	!DB/SMU moves through the available reports page by page in the order in which they are listed on the screen.
Type the option number for a report on the command line and press Enter.	!DB/SMU directly displays the selected report.
Type **N** on the command line	!DB/SMU displays the next report on the list and bypasses remaining pages of the current report.
Type **P** on the command line	!DB/SMU displays the previous report on the list and bypasses remaining pages of the current report.
Type **T** on the command line	!DB/SMU returns you to the Table Space Stats panel, the menu panel (or top panel in the hierarchy) for the scan reports.

Index Keys Summary Report

Overview

This unit describes the index keys summary report.

Purpose of the index keys summary report

This report summarizes information about the index. Use the index keys summary report to do the following:

- Obtain summary information about the number of
  - Records in the index
  - Keys in the index
  - Leaf pages in the index
  - Levels in the index
- Identify unique or non-unique keys
- Determine the average number of records per key and page
- Determine the percentage of pages used
- Determine the percentage of free space used
- Evaluate the need to restructure the key
- Evaluate DASD utilization
- Evaluate the need to reorganize the index space or table space

Background about the structure of a DB2 index

A DB2 index is a balanced tree. You should understand these characteristics about its levels in order to intelligently manage your spaces:

- The top layer contains the root page.
- The intermediate layers contain non-leaf pages.
- The bottom layer contains the leaf pages.
- Once any free space is consumed, additional insert processing tends to increase the number of levels by splitting the leaf pages.

Background about the number of index levels

The number of levels in an index increases when additions occur in the following:

- Key length
- Number of indexed rows
- Number of subpages selected (for type 1 indexes)
- Free space and free page values

Accessing the index keys summary report

To access the index keys summary report, select option 1 on the Index Statistics panel.

Elements of the index keys summary report

```
INDEX STATISTICS ------------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===>                                                 Scroll ===> PAGE
                INDEXSPACE DSNATX03 ON mm/dd/yy at 12:37  1

       1. INDEX KEYS SUMMARY

 2          KEY LENGTH        8             NON-UNIQUE KEYS      8

 3      RECORDS INDEXED  382,824
 4        DISTINCT KEYS  205,114             1.9  AVG RECORDS PER KEY        9
 5           LEAF PAGES    2,056          186.2  AVG RECORDS PER PAGE       10

 6  LEVELS IN THE INDEX        3
 7 ESTIMATED AFTER REORG       3                      PCT PAGES USED = 94.1  11

                                                      PCT FREESPACE  = 46.9  12

--------------------------------------------------------------------------------
 No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
  1. Keys  2. Dists  3. Pages  4. Usage  5. Freespace  6. Subpages  7. Levels
```

1 If this were a type 2 index, the heading line would read:

TYPE 2 INDEX DSNATX03 ON *mm/dd/yy* AT 12:37

2 Sum of the lengths of all key columns, plus 2 for each VARCHAR column (which are counted at full length), plus 1 for each nullable column

3 Number of indexed records

4 Number of distinct keys

5 Number of leaf pages

6 Number of levels currently in the index

7 Number of levels estimated for this index if an immediate reorganization occurred after collecting statistics (based on the free space and free page specifications at that time). Variable length rows may skew this value.

Elements of the index keys summary report (continued)

8 Key type:

- Non-Unique
- Unique
- Unique primary (Indicates the primary index in a relationship involving referential integrity.)

9 Average number of records per key (For unique indexes, this number should be 1.0.)

10 Average number of records per page

11 Percentage of used pages in the index

12 Percentage of space in the entire index space that is free, either as unused pages or as free space in used pages

Interpreting the index keys summary report

Use the index keys summary report to assist you in making these decisions:

- When to run the DB2 REORG utility

  Run the DB2 REORG utility if the number of levels needed after reorganization is less than the current number of levels in the index. (Refer to "Index Levels Summary Report" on page 238 for details on index levels.)

- When to restructure the index

  Consider restructuring the key if the average number of records per key is high. This is especially true for a clustering index.

- When to reallocate an index space

  Frequently, the index only uses a small percentage of its allocated space. Use !DB/SMU to identify those index spaces that can be reallocated to reclaim wasted DASD. Refer to the unit "Pages Analysis Report" on page 226 for the number of pages required by the index space.

Interpreting the Pct Pages Used field

The following chart shows how the Pct Pages Used field reflects DASD use.

Pct Pages Used Value	Indicates . . .
Large	Space allocated to the index is well used
Very large	Index will soon go into additional extents
Low	Poor DASD utilization (Most of the space is not being used.)

Reports for Index Keys Distribution

Overview

This unit and the next three provide information about the three reports for index keys distribution.

Purpose of the index keys distribution reports

During a scan, !DB/SMU calculates the number of additional keys that can fit in the available free space in each subpage and generates reports to show the following:

- How full the index pages are
- Location of space in the index pages
- Support for non-unique keys
- Distribution of records per key
- The 10 most frequently occurring keys

Distribution of Pages with Room for Keys

Overview

This unit describes the first index keys distribution report, the distribution of pages with room for keys.

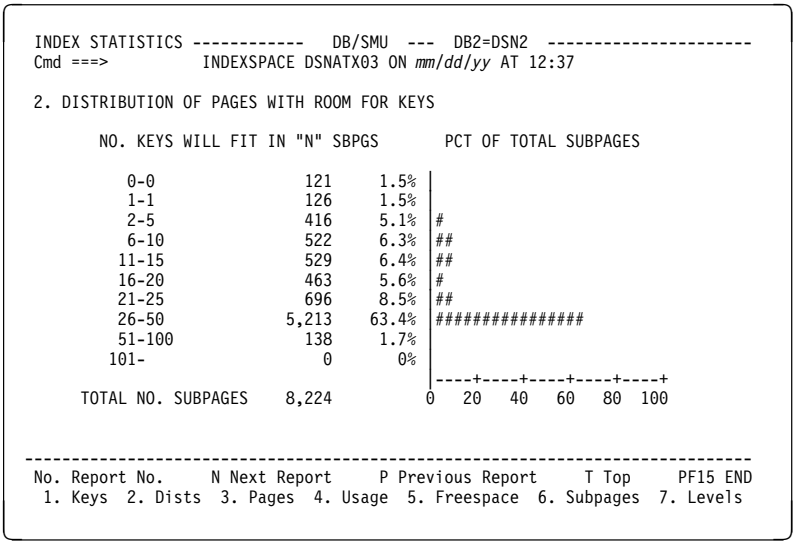
Accessing the report on the distribution of pages with room for keys

To access the first index keys distribution report, distribution of pages with room for keys, select option 2 on the Index Statistics panel.

Purpose of the distribution of pages with room for keys report

This report contains statistics and a histogram that show how full the pages of an index are.

Elements of the distribution of pages with room for keys report (type 1 index)



```
INDEX STATISTICS ------------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===>          INDEXSPACE DSNATX03 ON mm/dd/yy AT 12:37

2. DISTRIBUTION OF PAGES WITH ROOM FOR KEYS

       NO. KEYS WILL FIT IN "N" SBPGS       PCT OF TOTAL SUBPAGES

          0-0                121     1.5% |
          1-1                126     1.5% |
          2-5                416     5.1% |#
          6-10               522     6.3% |##
         11-15               529     6.4% |##
         16-20               463     5.6% |#
         21-25               696     8.5% |##
         26-50             5,213    63.4% |################
         51-100              138     1.7% |
        101-                   0       0% |
                                          |----+----+----+----+----+
     TOTAL NO. SUBPAGES    8,224           0   20   40   60   80  100


------------------------------------------------------------------------------
 No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
  1. Keys  2. Dists  3. Pages  4. Usage  5. Freespace  6. Subpages  7. Levels
```

Interpreting the distribution of pages with room for keys report (type 1 index)

The example report on the distribution of pages with room for keys shows the following:

- The sample index contains 8,224 subpages
- 121 subpages (1.5%) are full and have room for no keys
- 126 subpages (1.5%) have room for only 1 more key
- 5,213 subpages (63.4%) have room for 26 to 50 more keys

Therefore, this index has ample room to accommodate additional keys.

Elements of the distribution of pages with room for keys report (type 2 index)

```
INDEX STATISTICS ------------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===>          TYPE 2 INDEX DSNATX03 ON  mm/dd/yy AT 00:20

2. DISTRIBUTION OF PAGES WITH ROOM FOR KEYS

       NO. KEYS WILL FIT IN "N" PGS         PCT OF TOTAL PAGES

          0-0                  0          |
          1-1                  4    36.4% |#########
          2-5                  1     9.1% |##
          6-10                 0          |
         11-15                 2    18.2% |#####
         16-20                 3    27.3% |#######
         21-25                 1     9.1% |##
         26-50                 0          |
         51-100                0          |
        101-                   0       0% |
                                          |----+----+----+----+----+
     TOTAL NO. PAGES          11           0   20   40   60   80  100


------------------------------------------------------------------------------
 No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
  1. Keys  2. Dists  3. Pages  4. Usage  5. Freespace  6. Subpages  7. Levels
```

Interpreting the distribution of pages with room for keys report (type 2 index)

The example report on the distribution of pages with room for keys shows the following:

- The sample index contains 11 pages
- 4 pages (36.4%) have room for only 1 more key
- 1 page (9.1%) has room for 2 to 5 keys
- 2 pages (18.2%) have room for 11 to 15 keys
- 3 pages (27.3%) have room for 16 to 20 keys
- 1 page (9.1%) has room for 21 to 25 keys

This index may need reorganization depending on how often new keys are added to the space.

Distribution of Records per Key Report

Overview

This unit describes the second of the index keys reports, distribution of records per key.

Accessing the report on the distribution of records per key

Press Enter on the distribution of pages with room for keys report to access the report on the distribution of records per key. *This report is shown only if the index supports non-unique keys.*

Purpose of the distribution of records per key report

The following sample report contains statistics and a histogram that describe the number of records per key.

Elements of the distribution of records per key report

```
INDEX STATISTICS ------------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===>                 INDEXSPACE DSNATX03 ON  mm/dd/yy AT 12:37

2. B.  DISTRIBUTION OF RECORDS PER KEY

       NO. RECS INDEXED BY "N" KEYS         PCT OF TOTAL KEYS

          0-0                  0       0%  |
          1-1            170,718    83.2%  ####################
          2-5             26,853    13.1%  ###
          6-10             4,069     2.0%
         11-50             3,243     1.6%
        51-100                82       0%
       101-250                58       0%
       251-500                74       0%
       501-1000               8        0%
       1001-                  9        0%
                                          |----+----+----+----+----+
     TOTAL NO. KEYS      205,114          0   20   40   60   80  100


-------------------------------------------------------------------------------
 No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
  1. Keys  2. Dists  3. Pages  4. Usage  5. Freespace  6. Subpages  7. Levels
```

Note: If this report were for a type 2 index, the heading line would read: **TYPE 2 INDEX DSNATX03 ON *mm/dd/yy* AT 12:37**.

Interpreting the records per key report

The example report on records per key shows the following:

- The index contains a total of 205,114 keys
- 170,718 keys or 83.2% are unique, indexing only 1 record each
- 26,853 keys or 13.14% index from 2–5 records each
- 4,069 keys or 2% index from 6–10 records each
- 9 keys index over 1000 records each

Top Ten Records per Key Report

Overview

This unit describes the third of the index keys distribution reports, the top ten records per key report.

Accessing the report on the top ten records per key

Press Enter on the distribution of records per key report to access a report showing the top ten records per key. *It is shown only if the index supports non-unique keys.*

Purpose of the top ten records per key report

This report indentifies those ten keys that index the greatest number of records.

Elements of the top ten records per key report

The following report describes the 10 most-used keys.

```
INDEX STATISTICS ------------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===>                                                 Scroll ===> PAGE
               INDEXSPACE DSNATX03 ON  mm/dd/yy AT 12:37

2. C. TEN MOST FREQUENT KEYS

            NO. RECS    KEY VALUE           (FIRST 16 BYTES)
             INDEXED    CHARACTER           HEX     4       8       C
             -------    ----------------    --------------------------------
               2,270    ........            800880FD80000880
               2,062    ........            800580FD80000880
               1,977    ......c.            800480FD8000833B
               1,886    ......c.            800780FD8000833B
               1,471    ..a...a.            800281018000818B
               1,467    ........            800480FD80000880
               1,415    ..a...a.            800581018000818B
               1,215    ..a...a.            800481018000818B
               1,079    ..a...a.            800381018000818B
                 740    ..a...a.            8002810280000818B


No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
 1. Keys  2. Dists  3. Pages  4. Usage  5. Freespace  6. Subpages  7. Levels
```

Note: If this report were for a type 2 index, the heading line would read: **TYPE 2 INDEX DSNATX03 ON *mm/dd/yy* AT 12:37**.

Interpreting the most-used keys report

The example report on the most-used keys shows the following:

- Number of records each key indexes
- Character value for each key (The keys in this example are a composite of four SMALLINT columns, so these values are not meaningful.)
- Hexadecimal values of the first 16 bytes of each key

This example shows that each of nine keys index more than 1000 records. It also shows a fairly even distribution of records per key.

Interpreting the Three Index Keys Distribution Reports

Overview

This unit provides guidance in interpreting the information provided by the three index keys distribution reports.

Keys reports and the need to reorganize an index

The index keys reports help you evaluate the need to reorganize the index. Consider the following indicators:

- Numerous pages or subpages with room imply that there is less likelihood of additional page splits or of new index levels forming.
- Full pages or subpages imply that index levels are likely to split if new keys are inserted.

Use of key distribution information by the DB2 optimizer

During the access path selection process, the DB2 optimizer may attempt to select those indexes whose distribution of records per key shows few duplicates. That is, most of the keys appear in the 1-1 or 2-5 row. The optimizer will often place emphasis on the uniqueness of index keys.

Use of the top ten records per key report

Members of the applications development staff in particular find this report useful. Because of their knowledge of the business function of a particular data base, they can derive useful information about the table data and the appropriateness of the key structure from this report.

Problems related to multilevel indexes

An index in ascending-key sequence that grows from splits tends to

- Contain 50% free space
- Perform badly
- Require reorganization

Pages Analysis Report

Overview

This unit provides information about the pages analysis report.

Purpose of the pages analysis report

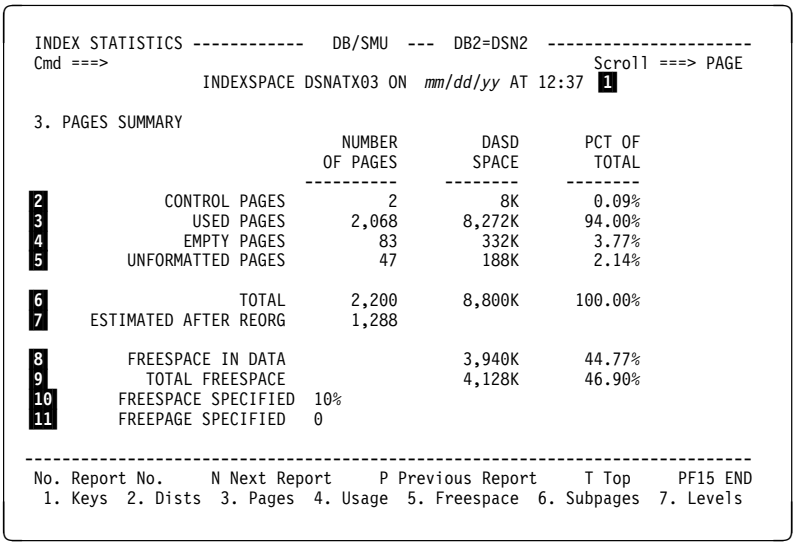
This report summarizes the status of all the pages in the index space. Use the pages anlysis report to do the following:

- Obtain page-use statistics for all the pages in the index space
- Determine how many pages that reorganization would save
- Obtain an adjustment value for efficient DASD use
- Obtain a summary on free space in the used pages and unformatted pages

Accessing the pages analysis report

To access the pages analysis report, select option 3 on the Index Statistics panel.

Elements of the pages analysis report



```
INDEX STATISTICS ------------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===>                                                  Scroll ===> PAGE
                  INDEXSPACE DSNATX03 ON  mm/dd/yy AT 12:37 1

3. PAGES SUMMARY
                                 NUMBER         DASD       PCT OF
                               OF PAGES        SPACE        TOTAL
                              ----------     --------     --------
2             CONTROL PAGES           2          8K        0.09%
3                USED PAGES       2,068       8,272K       94.00%
4               EMPTY PAGES          83         332K        3.77%
5         UNFORMATTED PAGES          47         188K        2.14%

6                     TOTAL       2,200       8,800K      100.00%
7     ESTIMATED AFTER REORG       1,288

8          FREESPACE IN DATA                  3,940K       44.77%
9            TOTAL FREESPACE                  4,128K       46.90%
10       FREESPACE SPECIFIED  10%
11       FREEPAGE SPECIFIED   0

------------------------------------------------------------------------------
 No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
  1. Keys  2. Dists  3. Pages  4. Usage  5. Freespace  6. Subpages  7. Levels
```

1 If this were a type 2 index, this heading line would read:

TYPE 2 INDEX DSNATX03 ON 11/3/99 AT 12:37

2 Number of DB2 control pages

3 Number of used pages

4 Number of empty formatted pages no longer holding data or unused formatted pages

5 Number of empty, unformatted pages never used by DB2

6 Totals for pages, DASD space, and the total percentage

7 Number of pages estimated for this index space if an immediate reorganization occurred after the scan (based on the free space and free page specifications at that time). Variable length rows may skew these results.

8 Percentage of free space in the used data pages

9 Percentage of total free space in the entire index space

10 Percentage of each leaf page (for type 2 indexes) or subpage (for type 1 indexes) DB2 is to leave free

11 Frequency of index pages DB2 is to leave empty

Interpreting the pages analysis report to optimize DASD use

To use DASD more efficiently, adjust the DASD space allocations for the index space to match the value in the Pages Needed After Reorg field as follows:

Type of Index Space	Action
For an index space in a storage group	Adjust the PRIQTY value in the index space definition.
For all other index spaces	Adjust the primary allocation value.

Usage Graph

Overview

This unit provides information on the Usage Graph.

Purpose of the usage graph

This graph summarizes the information about the entire DB2 index space into a single panel to show the distribution of used index pages in the dataset. Use this graph to do the following:

- Evaluate the need for reorganization
- Evaluate the need for reallocation

Background about the usage graph

The graph axes provide the following information:

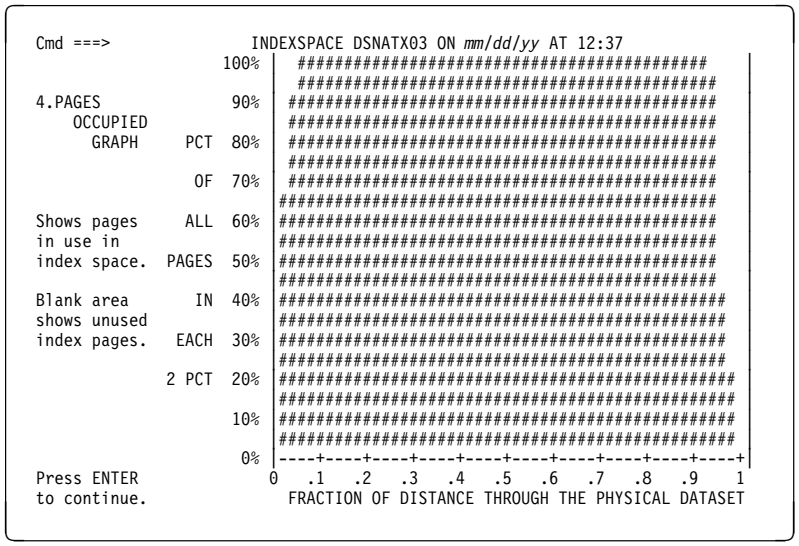
- Each of the grid marks at the bottom, labeled .1, .2, and so forth, represents 10% of the total index DASD space. Thus each dash (-) represents 2% of the total DASD space allocated to the index.
- Each column of the graph represents the percentage of occupied pages in that 2% of the index—2 cylinders if the index is 100 cylinders.

Accessing the usage graph

To access the usage graph, select option 4 on the Index Statistics panel.

Sample graph of occupied pages

This is a sample usage graph, that shows the occupied pages in the index.



Note: If this report were for a type 2 index, the heading would read: **TYPE 2 INDEX DSNATX03 ON *mm*/*dd*/*yy* AT 12:37**.

Interpreting the usage graph

Index spaces cannot be image copied, so this graph differs from kind of information provided by the table space update graph. The pattern of occupied pages often reveals index usage problems. The example usage graph shows the following:

- Almost all of the index pages are used. Therefore, reallocation into a larger space may be needed.

  See the next unit, “Free Space Graph” on page 231, to decide whether it is more appropriate to reallocate or to reorganize the space. In the example, the free space graph indicates that there is space available to be reclaimed. In this case, it is probably preferable to reorg rather than resize.

Free Space Graph

Overview

This unit provides information about the free space graph.

Purpose of the free space graph

This graph summarizes the entire DB2 index space into a single panel to show the distribution of free space in the index.

Background about the free space graph for indexes

The graph provides the following measurements:

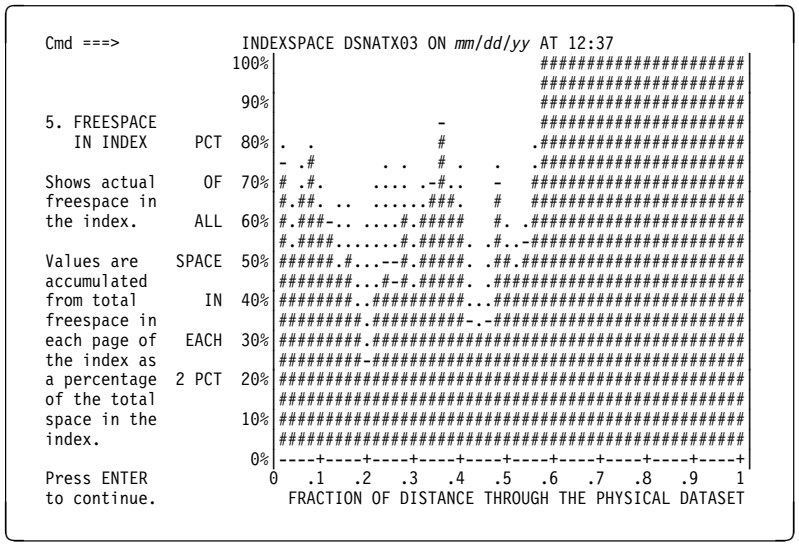
- Each of the grid marks at the bottom, labeled .1, .2, and so forth, represents 10% of the total index DASD space. Thus each dash (-) on the horizontal axis represents 2% of the total DASD space allocated to the index.
- Each column of the graph represents the percentage of occupied pages in that 2% of the index—2 cylinders if the index is 100 cylinders.
- Within the graph:
  - # represents a page with room for a long row
  - - represents a page with room for an average row
  - . represents a page with room for a short row

Accessing the free space graph for indexes

To access the free space graph for indexes, select option 5 on the Index Statistics panel.

Elements of the free space graph

This is a sample graph that shows the location and amount of free space in the index.



Note: If this report were for a type 2 index, the heading would read: **TYPE 2 INDEX DSNATX03 ON *mm/dd/yy* AT 12:37**.

Interpreting the need to reorganize the table space

A common pattern for the free space graph is one in which almost all pages are used but each is only about half full. This is caused by the index's splitting as new keys are added. While such an amount of free space provides for good insert performance for random inserts, it has no effect on sequential inserts and is harmful for retrievals—the main reason for establishing an index. In this instance, you should strongly consider reorganizing the table space.

Using the free space graph with the usage graph

The pattern of occupied pages often reveals index free space usage problems. The sample free space graph shows the following:

- Half of the space in the index is free.
- The index is growing by adding extents; this degrades performance.
- The index space should be reorganized.

Subpages Summary Report for Type 1 Indexes

Overview

This unit provides information about the subpages summary report.

Applicability of the subpages summary report

The subpages summary report is applicable to type 1 indexes only. A similar report, the leaf pages summary report, is available for type 2 indexes. See "Leaf Pages Summary Report for Type 2 Indexes" on page 235.

Purpose of the subpages summary report

Use the subpages summary report to obtain the following:

- Statistics for subpages
- Statistics for keys
- Statistics for records
- Size of each subpage

Background on index subpages

The following is true for index subpages:

- You can subdivide index pages into subpages when creating the type 1 index.
- Each physical page of the leaf-page level can be 1, 2, 4, 8, or 16 subpages.
- All subpages in a given index are the same size.
- The subpage is the unit of locking in the index.

Accessing the subpages summary report

Select option 6 on the Index Statistics panel to display the subpages summary report.

Elements of the subpages summary report

This is the subpages summary report.

```
INDEX STATISTICS ------------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===>                                                  Scroll ===> PAGE
                INDEXSPACE ISTSESUB ON  mm/dd/yy AT 00:20


6. SUBPAGES SUMMARY


 1   SUBPAGES PER PAGE         4
 2 SIZE OF EACH SUBPAGE       987

 3        DISTINCT KEYS   205,114          25.3  AVG KEYS PER SUBPAGE 4
 5     RECORDS INDEXED    382,824          47.3  AVG RECS PER SUBPAGE 6

 7   TOTAL NO. SUBPAGES     8,224
 8        USED SUBPAGES     8,095        98.43%
 9       EMPTY SUBPAGES       129         1.57%



--------------------------------------------------------------------------------
 No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
  1. Keys  2. Dists  3. Pages  4. Usage  5. Freespace  6. Subpages  7. Levels
```

1 Number of subpages per index page

2 Size of each subpage (Overhead in the physical index space makes the *actual* size of the subpage less than 4K divided by the number of subpages.)

3 Number of distinct keys in the index

4 Average number of keys per subpage

5 Number of records indexed

6 Average number of records per subpage

7 Total number of subpages

8 Quantity of used subpages

9 Quantity of empty subpages

Interpreting the subpages summary report

The use of subpages is a decision you can make when designing your database for optimum performance. The interpretation of this informational report is unique to your site.

Leaf Pages Summary Report for Type 2 Indexes

Overview

This unit provides information about the leaf pages summary report.

Applicability of the leaf pages summary report

The leaf pages summary report is applicable to type 2 indexes only. It is roughly equivalent to the subpages summary report for type 1 indexes. See "Subpages Summary Report for Type 1 Indexes" on page 233.

Purpose of the leaf pages summary report

Use the leaf pages summary report to obtain the following:

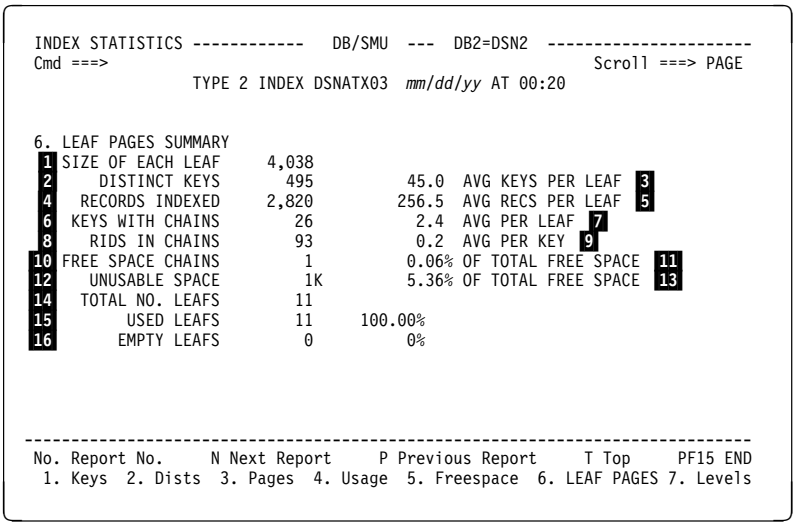
- Statistics for leaf pages
- Statistics for keys
- Statistics for records

Accessing the leaf pages summary report

Select option 6 on the Index Statistics panel for a type 2 index to display the leaf pages summary report.

Elements of the leaf pages summary report

This is the leaf pages summary report.



```
INDEX STATISTICS ------------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===>                                                  Scroll ===> PAGE
                TYPE 2 INDEX DSNATX03  mm/dd/yy AT 00:20


6. LEAF PAGES SUMMARY
 1 SIZE OF EACH LEAF     4,038
 2     DISTINCT KEYS       495          45.0  AVG KEYS PER LEAF  3
 4   RECORDS INDEXED     2,820         256.5  AVG RECS PER LEAF  5
 6  KEYS WITH CHAINS        26           2.4  AVG PER LEAF  7
 8    RIDS IN CHAINS        93           0.2  AVG PER KEY  9
10 FREE SPACE CHAINS         1          0.06% OF TOTAL FREE SPACE  11
12    UNUSABLE SPACE         1K         5.36% OF TOTAL FREE SPACE  13
14   TOTAL NO. LEAFS        11
15        USED LEAFS        11     100.00%
16       EMPTY LEAFS         0          0%




------------------------------------------------------------------------------
 No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
  1. Keys  2. Dists  3. Pages  4. Usage  5. Freespace  6. LEAF PAGES 7. Levels
```

1 Usable size of each leaf page

2 Number of distinct keys in the index

3 Average number of keys per leaf page

4 Number of records indexed

5 Average number of records per leaf page

6 Number of keys with RID (record identifier) chains

7 Average number of keys with RID chains per leaf

8 Number of RIDs in the chains

9 Average number of RIDs in chains per leaf

10 Number of free space chains

Elements of the leaf pages summary report (continued)

11 Percent of free space in chains as a percent of total free space

12 Unusable space

13 Unusable space as a percent of total free space

14 Total number of leaf pages in the space

15 Quantity of used leaf pages

16 Quantity of empty leaf pages

Interpreting the leaf pages summary report

Use the leaf pages summary report as an aid to understanding when to reorganize. You should consider reorganization when:

- There are many keys with RID chains
- There are many RIDs in a chain
- There is a large amount of unusable free space

Index Levels Summary Report

Overview

This unit provides information about the index levels summary report.

Purpose of the index levels summary report

This report analyzes key factors associated with index levels. Use the index levels summary report to do the following:

- Obtain summary statistics for the pages in each level
- Obtain summary statistics for the keys in each level
- Obtain summary statistics for the free space in each level
- Evaluate the possibility of an index split
- Evaluate the need to reorganize the index space
- Determine to what extent the physical table space order matches the index space logical order

Background about index levels

The following characteristics about index levels are important when managing space:

- DB2 indexes are balanced trees.
- The root page—the third page of the index—is always a single page in the highest level of the index.
- Each entry in the root page points to a page in the next level.
- For a type 1 index, the root page may also be a leaf page if all keys will fit onto one page.
- This tree-like structure continues until each entry in an index page points to a record in the indexed table space.

  (The number of keys at one level should always equal the number of pages at the next lower level.)
- The leaf page level is the lowest level in the index.
- Each level represents a potential I/O operation to retrieve a record from the table space.

Accessing the index levels summary report

To access the index levels summary report, select option 7 on the Index Statistics panel.

Elements of the index levels summary report

This is an index levels summary report.

```
 INDEX STATISTICS -----------   DB/SMU  ---  DB2=DSN2  ----------------------
 Cmd ===>                                                    Scroll ===> PAGE
                 INDEXSPACE DSNATX03 ON  mm/dd/yy at 12:37 [1]

 7. INDEX LEVELS SUMMARY
[2]   [3]         [4]      [5]       [6]        [7]        [8]       [9]
IX   PAGES IN     KEYS IN   KEYS  KEYS HAVE            AVG DIST  AVG DIST
LVL    LEVEL        LEVEL  /PAGE   ROOM FOR  FREESPACE  IN PAGES   IN TRKS
---  ---------  ----------- ------ -----------  --------  --------  --------
  1         1           11   11.0         357       99%

  2        11        2,056  186.9       1,992        4%     166.9      16.7

LEAF    2,056      205,114   99.8     198,713       53%      32.2       3.2

            [10] CLUSTERRATIO   100        CLUSTERING INDEX


******************************  END  OF  REPORTS  ***************************
No. Report No.     N Next Report     P Previous Report     T Top     PF15 END
 1. Keys  2. Dists  3. Pages  4. Usage  5. Freespace  6. Subpages  7. Levels
```

1 If this is a type 2 index, the heading line reads as follows:

TYPE 2 INDEX DSNATX03 ON mm/dd/yy AT 12:37

2 Number of the index level

3 Number of pages in the index level

4 Number of keys in the index level. (The number of keys at one level should always equal the number of pages at the next lower level.)

5 Average number of keys per page

6 Number of additional keys the level can support (Indicates the likelihood of an index level split.)

7 Percentage of free space in the level

8 Average number of pages between pages in the level—for the leaf level, the value is computed from logically consecutive pages. (Indicates the need for index space reorganization)

9 Average number of tracks between pages in the level—for the leaf level, the value is computed from logically consecutive pages. (Indicates the need for index space reorganization)

10 Measurement of how well the physical ordering of the table space matches the logical ordering of the keys in its index. A value of 100 indicates a perfect match.

Interpreting the index levels summary report

Review the Keys Have Room For field to help you assess when the index will go to an extra level. This is important since each level of an index represents a potential I/O operation.

Review the Average Distance in Pages and Average Distance in Tracks columns. A large number indicates the need for index space reorganization.

The sample report shows that the index space should be reorganized because of the following factors:

- The large Avg Dist in Pages value (32.2) indicates that 32.2 pages physically separate each leaf page. A sequential scan of this index would be extremely expensive in terms of I/O operations. Reorganizing the index would change this value to 1.
- Logically successive pages of this index are more than 3 tracks apart. This reduces the effectiveness of using a look-aside buffering scheme to reduce I/O.

Interpreting the CLUSTERRATIO value

The index levels summary report contains a CLUSTERRATIO value. The following chart explains fundamentals for using the CLUSTERRATIO value.

CLUSTERRATIO Factor	Explanation
Only one index for a table can be *declared* as clustering.	If none is declared, *clustering by default* occurs for the first index created for the table.
A high value indicates that the table rows are well ordered with respect to this index.	A logically sequential scan of the table should have few random or repeated table space I/Os. The DB2 optimizer favors indexes with a high CLUSTERRATIO value.
CLUSTERRATIO can be affected only by reorganizing the *table space.*	The value improves only for the clustering index and—unless the table space is segmented—only for one of the tables in the table space. To achieve this improvement, the table must be unloaded, reloaded in seqeuence, and sorted in clustering order.

Chapter 12. Using RUNSTATS Statistics Generated by !DB/SMU

Introduction

During a full space scan !DB/SMU calculates statistics that are similar to those gathered by the DB2 RUNSTATS utility. The scan statistics, however, offer certain additional advantages. This chapter provides information on using the !DB/SMU RUNSTATS statistics.

Chapter Contents

Overview of the RUNSTATS Statistics Generated by !DB/SMU

Overview

This unit provides an overview of the !DB/SMU RUNSTATS statistics.

Features of the !DB/SMU RUNSTATS

This list identifies the major features of the !DB/SMU RUNSTATS.

- !DB/SMU collects the statistics during a scan and timestamps them with a date and time. By default, the statistics are collected automatically; however, you can specify that statistics collection be turned off.
- If you request it, !DB/SMU generates the required SQL statements needed to update the DB2 catalog and saves them as member *userid*R in the user PDS. The statments optionally can be saved in the log PDS.
- The RUNSTATS reports are retained until the number of generations of statistics specified by the value for the Max Reports option in the System Options is reached.

What statistics are displayed?

!DB/SMU RUNSTATS statistics vary according to the type of data space they were generated for. This chart provides information about the type of information provided for a given type of space.

	Nonpartitioned Table Space	Partitioned Table Space	Nonpartitioned Index Space	Partitioned Index Space
Individual statistics for each table	√			
Summary statistics for the entire space	√		√	
Partition statistics		√		√
Aggregate statistics for all partitions		√		√

Displaying Online RUNSTATS Statistics

Overview

This unit describes how to display the !DB/SMU RUNSTATS information.

Accessing RUNSTATS

You access the available RUNSTATS statistics calculated by !DB/SMU from the Report Select List panel.

Effect of using the RUNSTATS command or the R select to display statistics

Task You Want to Perform	Action to Take
Replace the list of reports on the Report Select List panel with a list of RUNSTAT-type statistics !DB/SMU produces.	Type **RUNSTATS** on the command line of the Report Select List panel and press Enter. **Result:** !DB/SMU replaces the list of reports on the panel with a list of summary RUNSTATS-type statistics.
Expand the summary line of RUNSTATS statistics on the Report Select List panel to show a full panel of catalog columns updated with the information generated by the !DB/SMU RUNSTATS utility.	Type **S** in the Sel field of the Report Select List panel next to a line of summary RUNSTATS you are interested in.
Take a fast path to the expanded RUNSTATS full-panel display.	Type **R** in the Sel field of the Report Select List panel next to a scan report for which you want to see the expanded RUNSTATS.

Updating the DB2 Catalog with RUNSTATS Statistics

Overview

This unit contains procedures for updating the DB2 catalog with the RUNSTATS statistics that are generated during a scan.

Prerequisites for updating the DB2 catalog

You must have authority to update the DB2 catalog tables to complete this procedure.

Background about using the online interface for RUNSTATS

You can use the RUNSTATS online panels to automatically generate SQL statements to update the DB2 catalog. (For partitioned table spaces, !DB/SMU also generates SQL statements that reflect aggregate statistics.) Using the SQL statements it has generated, !DB/SMU updates the DB2 catalog tables at your command so that DB2 can use the new values at bind time.

Generating SQL update statements for RUNSTATS statistics

Follow these steps to generate SQL statements from the online RUNSTATS displays.

Step	Action
1	Access the table space or index space RUNSTATS display.
2	Perform the appropriate action. • To generate standard update statements for a table space or index space, type **GEN** in the command field. • To generate aggregate update statements for a partitioned table space, type **GENA** in the command field.
3	Press Enter. **Result**: !DB/SMU saves the statements in the user PDS in the member *userid*R.

Sample update statements for RUNSTATS for a nonpartitioned table space

The following figure contains sample update statements for a nonpartitioned table space. The GEN command generated these statements from an online RUNSTATS display.

```
EDIT ---- TDKT.DBTOOLS.V31.DB31.SMU.USER(USERIDR) - 01.00 ---- COLUMNS 001 072
COMMAND ===>                                                 SCROLL ===> PAGE
****** ***************************** TOP OF DATA ******************************
000001     UPDATE SYSIBM.SYSTABSTATS
000002        SET
000003            CARD               = 1                                    ,
000004            NPAGES             = 1                                    ,
000005            PCTPAGES           = 8                                    ,
000006            NACTIVE            = 12                                   ,
000007            PCTROWCOMP         = 0                                    ,
000008            STATSTIME          = CURRENT TIMESTAMP
000009      WHERE DBNAME             = 'ZETTI'
000010        AND TSNAME             = 'CUSTOMER'
000011        AND PARTITION          = 2
000012 ;
****** **************************** BOTTOM OF DATA ****************************
```

Sample update statements for aggregate RUNSTATS for a partitioned table space

The following figure contains sample update statements for aggregate RUNSTATS statistics for a partitioned table space. The GENA command generated these statements from an online RUNSTATS display.

```
EDIT ---- TDKT.DBTOOLS.V31.DB31.SMU.USER(USERIDR) - 01.00 ---- COLUMNS 001 072
COMMAND ===>                                                 SCROLL ===> PAGE
****** ***************************** TOP OF DATA ******************************
000001     UPDATE SYSIBM.SYSTABLESPACE
000002        SET
000003            NACTIVE            = 312                                  ,
000004            STATSTIME          = CURRENT TIMESTAMP
000005      WHERE DBNAME             = 'ZETTI'
000006        AND NAME               = 'CUSTOMER'
000007 ;
000008     UPDATE SYSIBM.SYSTABSTATS
000009        SET
000010            CARD               = 1                                    ,
000011            NPAGES             = 1                                    ,
000012            PCTPAGES           = 8                                    ,
000013            NACTIVE            = 12                                   ,
000014            PCTROWCOMP         = 0                                    ,
000015            STATSTIME          = CURRENT TIMESTAMP
000016      WHERE DBNAME             = 'ZETTI'
000017        AND TSNAME             = 'CUSTOMER'
000018        AND PARTITION          = 2
000019 ;
000020     UPDATE SYSIBM.SYSTABLES
000021        SET
000022            CARD               = 26                                   ,
000023            NPAGES             = 26                                   ,
000024            PCTPAGES           =  8                                   ,
000025            PCTROWCOMP         =  0                                   ,
000026            STATSTIME          = CURRENT TIMESTAMP
000027      WHERE CREATOR            = 'TSGM57'
000028        AND NAME               = 'CUSTOMER'
000029 ;
****** **************************** BOTTOM OF DATA ****************************
```

Sample update statements for RUNSTATS for a nonpartitioned index space

The following figure contains sample update statements for a nonpartitioned index space. The statistics that are generated are summary statistics for the entire space. The GEN command generated these statements from an online RUNSTATS display. Since !DB/SMU does not collect statistics information for FIRSTKEYCARD, NEAROFFPOS, and FAROFFPOS, if they appear in the report, their value is set to -1.

```
EDIT ---- TDKT.DBTOOLS.V31.DB31.SMU.USER(USERIDR) - 01.00 ---- COLUMNS 001 072
COMMAND ===>                                                 SCROLL ===> PAGE
****** **************************** TOP OF DATA ******************************
000001     UPDATE SYSIBM.SYSINDEXES
_
000002        SET
000003            CLUSTERRATIO       = 100                                  ,
000004            FIRSTKEYCARD       = 1                                    ,
000005            FULLKEYCARD        = 0                                    ,
000006            NLEAF              = 1                                    ,
000007            NLEVELS            = 1                                    ,
000008            STATSTIME          = CURRENT TIMESTAMP
000009      WHERE CREATOR            = 'TSBQQQ'
000010        AND NAME               = 'XACT1'
000011 ;
****** **************************** BOTTOM OF DATA ****************************
```

Sample update statements for RUNSTATS for a partitioned index space

The following figure contains sample update statements for a partitioned index space. The statistics that are generated are for the partition. The GENA command generated these statements from an online RUNSTATS display. Since !DB/SMU does not collect statistics information for FIRSTKEYCARD, NEAROFFPOS, and FAROFFPOS, if they appear in the report, their value is set to -1.

```
EDIT ---- TDKT.DBTOOLS.V31.DB31.SMU.USER(USERIDR) - 01.00 ---- COLUMNS 001 072
COMMAND ===>                                                 SCROLL ===> PAGE
****** **************************** TOP OF DATA ******************************
000001     UPDATE SYSIBM.SYSINDEXSTATS
000002        SET
000003            FIRSTKEYCARD       = 1                                    ,
000004            FULLKEYCARD        = 1                                    ,
000005            NLEAF              = 1                                    ,
000006            NLEVELS            = 1                                    ,
000007            CLUSTERRATIO       = 100                                  ,
000008            STATSTIME          = CURRENT TIMESTAMP
000009      WHERE OWNER              = 'TSGM57'
000010        AND NAME               = 'CUST'
000011        AND PARTITION          = 1
000012 ;
****** **************************** BOTTOM OF DATA ****************************
```

Turning Off RUNSTATS Statistics Collection

Overview

This unit describes the process for turning off statistics collection.

Rationale for omitting statistics collection

By default, !DB/SMU collects RUNSTATS information as a byproduct of a scan. When the scan involves partitioned spaces, !DB/SMU provides aggregate statistics for the space. The accumulation of statistics, particularly aggregate statistics, can be costly in terms of system resources, particularly when the majority of the spaces in your subsystem are partitioned. Turning off the accumulation of aggregate RUNSTATS, or turning off RUNSTATS information totally, can save system resources.

Procedure for omitting statistics collection

!DB/SMU provides four commands that permit you to selectively gather or omit the gathering of RUNSTATS. These are explained in the chart.

Command	Meaning
RUNSTATS (default)	Collect RUNSTATS information
NORUNSTATS	Do not collect RUNSTATS information
AGGSTATS (default)	Collect aggregate RUNSTATS information for partitioned spaces
NOAGGSTATS	Do not collect aggregate RUNSTATS information for partitioned spaces

Using the commands

To use the RUNSTATS commands, follow the steps in this chart.

Step	Action
1	Generate the JCL for a batch scan run following the instructions in “Generating Batch Jobs” on page 469.
2	When !DB/SMU displays the Process Member panel, select option 2, Edit.

3	Edit the batch JCL to add one of the RUNSTATS commands. The command should be entered in the JCL following the SYSIN DD statement and ahead of the SCAN *profilename* command. Here is an example:
	.
	.
	.
	`//SYSTSIN DD *`
	` ISPSTART CMD(%KTSXSMU DB2(DB31)) NEWAPPL(KTC1)`
	`//SYSIN DD *`
	` NOAGGSTATS <=== Inserted Command`
	` SCAN TSTR <=== Scan Command`
	`/*`
	`//`

4	Once editing is complete, type END on the command line or press the appropriate function key to return to the Process Member panel.
5	Submit your batch job by selecting option 1, Update Now.

Conflict between your JCL generation options and an inserted command

On the Generate Batch Job panel, you are provided with the option to specify whether or not to update RUNSTATS by typing either **Y** or **N** in the Update RUNSTATS field. If you should specify Update RUNSTATS===> Y on the Generate Batch Job panel, and then insert a NORUNSTATS or NOAGGSTATS command in your batch JCL, you have created a conflict. In this case, !DB/SMU *will* collect RUNSTATS and perform the update as you requested on the Generate Batch Job panel. Additionally, !DB/SMU will issue a message on the batch summary report indicating that RUNSTATS have been collected and the update performed.

Chapter 13. !DB/SMU Data Integrity Checks

Introduction

This chapter provides information about the integrity checks !DB/SMU makes on table spaces and index spaces.

Chapter Contents

Chapter Overview

Overview

This chapter describes the integrity checks !DB/SMU performs. These checks are described by online help, but you may find it convenient to refer to this chapter when you are correcting data integrity errors.

Organization of this chapter

The chapter covers the types of integrity checks that !DB/SMU makes to each DB2 table space and index space that it scans. This chart categorizes the integrity checks and identifies the unit in this chapter where you can find additional information.

Type of Space	Checks !DB/SMU Performs	Detailed Information
Table spaces and index spaces	• Header page checks • Page header checks • Page-type checks	"Integrity Checks for Table Spaces and Index Spaces" on page 252
Simple and partitioned table spaces	• Data page checks • Data page record checks • Overflow rows • Dropped tables • Pointer records • Data page ID map checks	"Integrity Checks for Table Spaces" on page 253
Segmented table spaces	• Space map page checks • Data page checks	"Integrity Checks for Segmented Table Spaces" on page 255
Index spaces	• Checks on all pages • Logical page header checks • Cross-page checks • Leaf page chains	"Integrity Checks for Index Spaces" on page 256

Accessing and modifying the status of integrity checks

Information about accessing the list of available integrity checks and modifying their status can be found in “Enabling Data Integrity Checks” on page 131.

Error messages for integrity checks

The error messages associated with the various checks in this chapter are in “Error Messages for Integrity Checks” on page 657.

Source for related DB2 internals information

This chapter briefly refers to DB2 internals which are oriented to the external format of data pages. Refer to the IBM *DB2 Diagnosis Guide and Reference* for related information on this topic.

Integrity Checks for Table Spaces and Index Spaces

Overview

This unit discusses the integrity checks that !DB/SMU applies to both table spaces and index spaces during a scan.

Header page checks

The header page for each DB2 table space and index space contains values that identify the database and the space within that database. !DB/SMU checks this identification.

Page header checks

Each page in a DB2 space contains a prefix called a *header*. The length of the header is 20 bytes for table spaces; 28 bytes for index spaces. The header identifies the page by number and contains values that DB2 uses to manage the page.

!DB/SMU validates all page headers by reviewing these items:

- I/O error flag
- Page end flag
- Page number
- Broken page
- Update status (table space only)

Checks for page type

DB2 marks each page with a type flag, indicating what kind of page it is. There are *header* pages (first page), *space map* pages, *data* pages, *index* pages, *index root* pages, *index leaf* pages, and *unformatted* pages.

!DB/SMU checks that the page type is consistent with the type of DB2 dataset being processed and confirms that the page type is valid relative to the page number. (For example, space map pages occur at specific intervals.)

Integrity Checks for Table Spaces

Overview

This unit describes the integrity checks !DB/SMU performs for all table spaces during a scan.

Data page checks

!DB/SMU checks these values that DB2 maintains in the page header to describe the format of each data page:.

- Number of ID map entries (each 2 bytes at the end of the page)
- Offset to the beginning of the contiguous, free space

DB2 maintains free space in the page in non-contiguous pieces called *holes.* Large holes (more than 4 bytes) are chained from the header. Small holes (1–4 bytes) can be scattered throughout the page. The *contiguous free space,* which is always immediately in front of the ID map area, is not on the hole chain. The offset to this contiguous free space is in the header, as is the total amount of free space in the page. !DB/SMU validates the free space total value, the hole chain, and the contiguous free space.

Data page record checks

Each element in a DB2 page is identified by a unique identifying type flag. Elements can be *holes, records, overflow records*, or *pointers* to overflow records.

Each record also contains an *ID map entry* number in its prefix, the first 6 bytes of the record. The ID map area is a set of contiguous 2-byte fields. DB2 indexes a record to a page and a ID map entry number. This allows DB2 to freely rearrange the records in a page without needing to update any index. The ID map entry for a record contains the offset to the record within the page.

Overflow rows

An overflow record is one which has been updated and whose length has increased such that there was no longer room in the original page for the new, longer record. DB2 replaces the original record with a 6-byte pointer record to the new location of the record.

Before resorting to overflow records, DB2 compacts any fragmented, free space in a page in an attempt to create sufficient contiguous space for the new record. By definition, therefore, an overflow record must be in a different page from its pointer.

Dropped tables

Records and overflow records contain the DB2 internal table ID number (OBID). !DB/SMU verifies this record prefix value against the OBIDs of all tables defined in the table space when the catalog extract was performed. If a matching OBID is not found, the record is assumed to be in a dropped table and is accumulated as such in the various !DB/SMU statistics reports.

Pointer records

Each pointer record must point to an overflow record, and each overflow record must be pointed to by exactly one pointer record. !DB/SMU checks all these conditions and issues error messages as appropriate.

Data page ID map checks

The ID map in the DB2 data page contains the offset to the records in the page. DB2 indexes ID map entries, not records. It is critical that these entries be correct. Unused ID map entries are marked as free (X'80' flag on), and all the free entries are chained together. !DB/SMU checks this chain of free ID map entries. When processing the records themselves, !DB/SMU checks that the ID map entry for that record is valid. At the end of the page scan, !DB/SMU checks that all ID map entries are accounted for.

Integrity Checks for Segmented Table Spaces

Overview

This unit describes the integrity checks !DB/SMU performs for segmented table spaces during a scan:

Space map page checks

!DB/SMU ensures that

- Page type flags are valid.
- The recorded segment size matches the definition of the table space.
- Coverage of the page is valid for the page and segment size combination.
- The next segment to be allocated is correctly identified.
- Numbers of allocated and free segments and used pages are in accord.

!DB/SMU also checks each segment entry to ensure that the entry type is valid and that the OBID is valid for the type. Chains of segment entries are validated, and any errors detected are reported (for example, a broken chain, loop in chain, or unchained entries).

Data page checks

!DB/SMU checks each data page of a segmented table space to ensure that all the rows it contains are from the table identified in the corresponding segment entry in the preceding space map page.

In addition, all data pages are checked to ensure that they are within the range of pages *owned* by the preceding space map page and that they are flagged as pages from a segmented table space.

Integrity Checks for Index Spaces

Overview

This unit describes the integrity checks !DB/SMU makes for index spaces during a scan.

Checks on all pages

Each page in a DB2 index has a *physical page header* at the start of the page. This physical page header is 28 bytes in length for a type 1 index and 44 bytes in length for a type 2 index. It contains the page number, flags that identify the type of page, and other values that vary depending on the type of page.

For a type 1 index, at least one *logical page header* follows the physical page header. There are multiple logical page headers in segmented pages. Most of the checks made on the physical page header are the same as those for table spaces.

In the case of a type 2 index, the physical page header has an appendage, the nature of which depends on whether the page is a leaf page or nonleaf page.

Logical page header checks

There are 2 different kinds of used index pages— *leaf pages* and *nonleaf pages.* Leaf pages are index pages at the lowest level of the index. Each entry in a leaf page points to a record in the indexed table. Nonleaf pages are pages in the hierarchy of the index that point to other index pages. The index page at the top of the hierarchy is always page 2 (the third page), the *root page.*

Cross-page checks

!DB/SMU performs extensive testing on all pointers in the index hierarchy. The entire index structure is tested for validity. Each entry in the root page must point to a page in the second level of the index. Each entry in each page of the second level must point to a page in the third level of the index. This goes on until the lowest level of the index, the leaf page level, is reached. At the leaf page level, each entry points to a record in the table being indexed.

Additionally, at the leaf page level, each page points to its predecessor and successor, creating an ordered chain of all the leaf pages. !DB/SMU checks all these pointers within the index. Note, however, that pointers from the index to the table (that is, from the index space to the table space) are *not* checked.

Leaf page chains

The leaf pages in a DB2 index are chained in both directions. DB2 uses this chain when reading the leaf pages during a nonmatching index scan. !DB/SMU checks the integrity of this chain.

The Space Map Function

Chapter 14.
Using the Space Map Function

Introduction

This chapter contains information about using the space map analysis facility. It includes information on the reports, maps, and graphs provided by this facility.

Chapter Contents

Chapter Overview

Prerequisites for using this chapter

Before running the space map analysis function, do the following:

- Review the information on selecting spaces in “Identifying Spaces to Work With” on page 77
- Ensure that you have appropriate authority for !DB/SMU
- Ensure that you have appropriate authority for dataset access required by a control system such as RACF

Organization of information in this chapter

This chapter describes the space map analysis function and covers procedures for performing an analysis and displaying the results.

Online help and this chapter

This chapter contains descriptions of the online panels you use to perform a space map analysis and the procedures you follow. Online help provides an additional level of detailed information about the panels and their fields.

Overview of the Space Map Analysis Function

Overview

This unit describes the way in which the space map analysis function operates and lists the output of the analysis.

Background about a space map analysis

Running a space map analysis provides information that enables you to evaluate the status of a physical DB2 object, such as a table space or an index space, as it existed at a particular time. Running a space map analysis is an excellent way to quickly determine the status and content of a space.

Fast access to space map information

When performing a space map analysis, !DB/SMU reads only the space map pages contained in the DB2 space. !DB/SMU does not access the space to build the space graphs unless a new analysis is requested. !DB/SMU keeps the space data in the report member of its system PDS.

Since !DB/SMU performs so few I/Os to build its graphs, you can run new space map analyses more frequently to observe trends with little impact to system performance.

Background about space map pages

DB2 maintains space map pages in all table spaces and index spaces. These special pages are kept at regular intervals throughout the DB2 dataset. Each space map page owns the data pages that follow it.

Information provided by DB2 space map pages for table spaces

The DB2 space map pages for table spaces hold data that tells DB2 whether each of the owned pages has free space and how much, and which pages have been updated since the previous execution of the COPY utility.

How a space map analysis works

!DB/SMU provides a quick, qualitative analysis of DB2 space. During a !DB/SMU space map analysis

- The DB2 dataset is allocated and opened.
- !DB/SMU analyzes the space map pages.
- The dataset closed and deallocated.

Generations of space map reports

You specify how many generations of space map reports !DB/SMU is to maintain. When this limit is reached, !DB/SMU replaces the oldest generation by a new one. Even though a report may be deleted because the limit is reached, !DB/SMU keeps summary statistics. You can delete these summary statistics explicitly, or let them expire once their own controlling limit is reached. You can also delete unwanted generations of reports before their expiration limit is reached.

Reports produced by a space map analysis

The !DB/SMU space map analysis generates these reports:

- Reports for table spaces
  - Summary report
  - Space graphs
  - Update graphs
  - Free space maps
  - Update maps
- Reports for index spaces
  - Summary report
  - Usage graphs
  - Usage maps

Zooming

The most detailed format for a space map is 1 screen character for each page of the DB2 space. For large DB2 spaces, such a map would extend through many, many panels. To reduce the amount of scrolling required to review a space map, !DB/SMU provides a zoom facility to assist in pinpointing the area of the space under investigation.

Cylinder zoom

Initially, a !DB/SMU space map displays with the zoom set to cylinder. In this mode, each screen character represents a composite value for an entire cylinder of the DB2 space. To view and evaluate a specific area of the map, quickly scroll through the cylinder map; then explode the area of interest by changing the zoom setting to track.

Track zoom

When using track zoom, each screen character is a composite value for 1 track of the space. You can use ISPF scrolling to further localize the area to investigate, and change the field to page zoom.

Page zoom

When using page zoom, each screen character represents a single DB2 page. You can use the space map to determine the specific page numbers to investigate.

Units of measure for the vertical axis

Regardless of the current zoom (cylinder, track, or page), the vertical axis of a map is always in cylinder (CYL) units if the pages in the space are 4K pages, or in page (PAG) units if the pages in the space are 32K pages.

Example of zooming

This example shows a space map for a table space with the zoom set to cylinder.

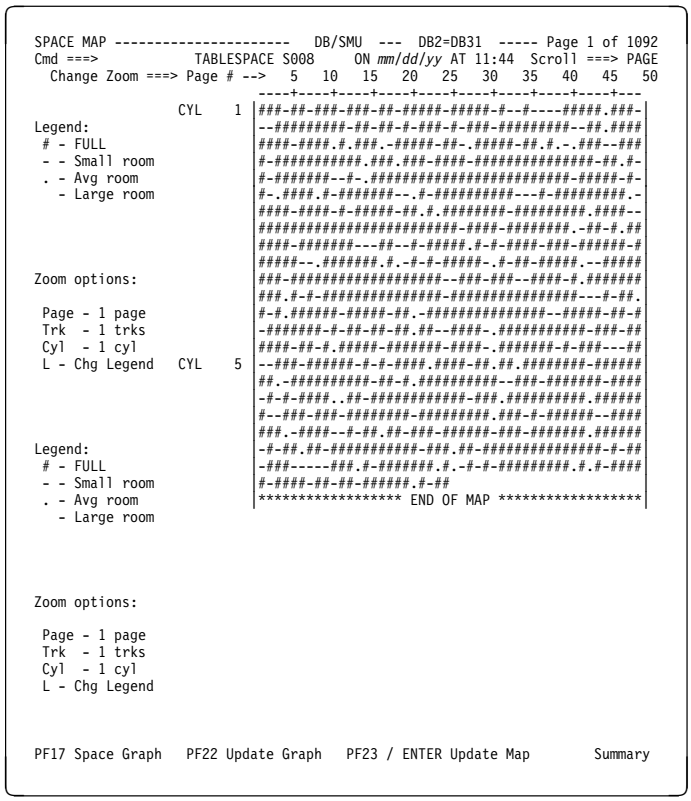
```
SPACE MAP ----------------------   DB/SMU  ---  DB2=DB31  --------- Cyl 1 of 7
Cmd ===>            TABLESPACE S008     ON mm/dd/yy AT 11:44  Scroll ===> PAGE
  Change Zoom ===> Cyl  # -->   5   10   15   20   25   30   35   40   45   50
                              ----+----+----+----+----+----+----+----+----+----+
                  CYL    1 |######                                             |
Legend:                     |****************** END OF MAP ******************|
 # - FULL
 - - Small room
 . - Avg room
   - Large room



Zoom options:

 Page - 1 page
 Trk  - 1 trks
 Cyl  - 1 cyl
 L - Chg Legend


PF17 Space Graph   PF22 Update Graph   PF23 / ENTER Update Map        Summary
```

To obtain greater detail for a particular area of the table space, change the zoom value from CYL to TRK by overtyping the value in the Change Zoom field.

```
SPACE MAP ----------------------   DB/SMU  ---  DB2=DB31  -------- Trk 1 of 91
Cmd ===>            TABLESPACE S008     ON mm/dd/yy AT 11:44  Scroll ===> PAGE
  Change Zoom ===> Trk  # -->   5   10   15   20   25   30   35   40   45   50
                               ----+----+----+----+----+----+----+----+----+----+
                  CYL    1 |##########-########################-#######|
Legend:           CYL    4 |####################-####################|
 # - FULL         CYL    7 |                                          |
 - - Small room             |*************** END OF MAP ****************|
 . - Avg room
   - Large room


Zoom options:

 Page - 1 page
 Trk  - 1 trks
 Cyl  - 1 cyl
 L - Chg Legend

PF17 Space Graph   PF22 Update Graph   PF23 / ENTER Update Map        Summary
```

Example of zooming (continued)

If zooming to a track view in not sufficient to understand the availability of room in the table space, change the zoom value to PAGE, again by overtyping the value in the Change Zoom field.



Space map analysis in batch

When you run the batch utility SPACE command, the reports named above are produced in batch. See “Defining and Using Run Profiles for Batch Jobs” on page 491 and “Using the Batch Reports for !DB/SMU” on page 591 for information about batch space map analysis.

Performing a Space Map Analysis

Overview

This unit contains procedures for performing an online space map analysis.

Background about beginning a space map analysis

You can access the panels to perform a space map analysis from two options on the !DB/SMU Primary Menu:

- Spacemaps

  Use this selection to obtain new reports or view existing ones. This option will be covered in the balance of this chapter.

- Monitor

  You can view existing reports using this selection. Use this selection to analyze sets or ranges of values so you can isolate problems. This option is covered in detail in "Using the Monitor Functions" on page 301.

Beginning a space map analysis

Review "Prerequisites for using this chapter" on page 262 before starting a space map analysis. Then follow the steps in this chart.

Step	Action
1	Access the !DB/SMU Primary Menu.
2	Type **2** in the Option field.
3	Press Enter. **Result**: !DB/SMU displays the Data Space Selection panel.
4	Complete the Data Space Selection panel by specifying the criteria identifying the spaces for which you want to see an analysis. (Refer to "Using the Data Space Selection Panel to Create a Selection List" on page 81 for details.)
5	Press Enter. **Result**: !DB/SMU displays the DB2 Space Map Analyses panel which lists the spaces meeting the criteria you specified.

Beginning a space map analysis (continued)

Step	Action
6	To request a new analysis, take one of these actions on the DB2 Space Map Analyses panel: • Type **N** in the Sel field next to a table space or index space, or • Type **S** in the Sel field next to a table space or index space and press Enter. Then type **S** in the "Select here ==> _ to perform a new full space scan" field of the Report Select List panel.
7	Press Enter. **Result**: !DB/SMU begins the space map analysis for the space or spaces you identified.

Chapter 15.
Space Map Analysis Reports for Table Spaces

Introduction

This chapter contains information about using the space map analysis reports for table spaces. It contains example reports, maps, and graphs.

Chapter Contents

Chapter Overview

Prerequisites for using this chapter

Before reviewing the information in this chapter, read the chapter “Using the Space Map Function” on page 261.

Organization of information in this chapter

The units in this chapter describe the table space reports generated by an online space map analysis. Each unit describes the way in which you access the report using the !DB/SMU online interface and describes the contents of the report.

Generating table space space map analysis reports in batch

You can also use batch procedures to generate similar reports. For information on the batch versions, see “Using the Batch Reports for !DB/SMU” on page 591.

Types of space map analysis reports for table spaces

The table space space map analysis reports provide details to assist you in improving the performance of your DB2 data bases. This chart provides summary information about each space map analysis report and points you to detailed information.

Table Space Report	Report Contents	Detailed Information
Space Summary	Numerical summary of free space and updates	"Space Summary Report" on page 276
Space Graph	Graph of the distribution of free space within the DB2 dataset	"Space Graph for Table Spaces" on page 279
Update Graph	Graph of the distribution of updated pages in the DB2 dataset available for rows	"Update Graph for Table Spaces" on page 281
Free Space Map	Detailed map showing the measure of free space in each page of the DB2 dataset	"Space Map for Table Spaces" on page 284
Update Map	Detailed map showing the measure of free space in each page of the DB2 dataset	"Update Map for Table Spaces" on page 286

Using Space Map Reports for Table Spaces

Overview

Each time you run the !DB/SMU space map analysis function, it creates a series of table space reports. This unit describes how to access the reports and navigate through them.

Selecting the appropriate generation of reports

When you access the Report Select List panel, you are presented with a list of all available reports for a table space, sorted in date order with the most recent generation of reports at the top of the list. You can view these most recent reports or earlier ones.

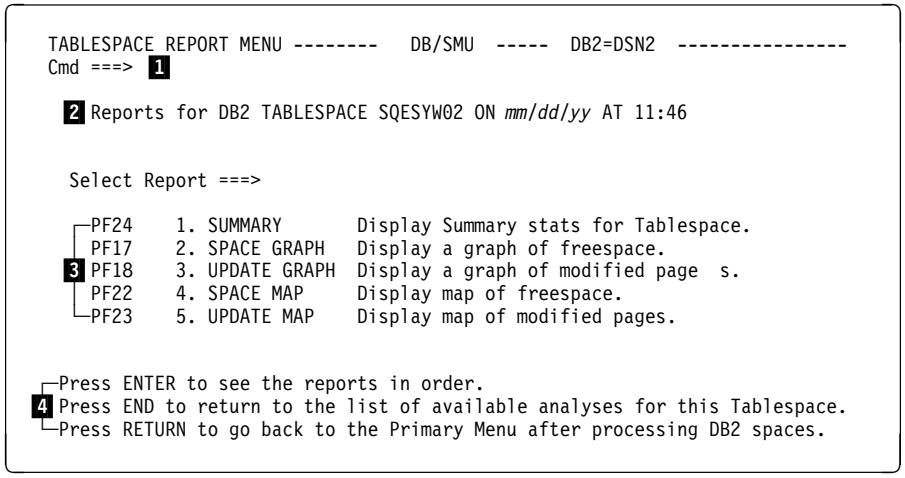
Accessing the space map analysis reports for table spaces

Follow these steps to access the space map analysis reports for table spaces.

Step	Action
1	Access the Report Select List panel as described in “Viewing a Report” on page 91.
2	In the list of available reports, type **S** in the SEL column next to the reports you want to see.
3	Press Enter. **Result:** !DB/SMU displays the Tablespace Report Menu panel.
4	On the Tablespace Report Menu panel, type the number of the report you want to see on the command line or press the appropriate PF key. If you want to view the reports in order, press Enter.

Elements of the Tablespace Report Menu panel

This is the Tablespace Report Menu panel.



1 Command line; type the number of the report you want to view here.

2 Name of table space for which the reports have been generated, and the date and time of this generation of reports

3 Basic set of available reports for table spaces

4 Additional actions you can perform on this panel. If you have selected multiple reports on the Report Select List panel, !DB/SMU displays all of the reports before executing the END or RETURN commands.

Space Summary Report

Overview

This unit describes the summary report for table spaces.

Purpose of the summary report for table spaces

The summary report for table spaces provides summary information about the use of space within the table space as well as the frequency and distribution of updates within the space.

Accessing the summary report for table spaces

To access this report, select option 1 on the Tablespace Report Menu panel.

Elements of the table space summary report

This is the table space summary report.

```
TABLESPACE SUMMARY -------------   DB/SMU  ---  DB2=DB31  ---------------------
Cmd ===>                                                           yy/mm/dd 08:34

                  TABLESPACE CHECKTS  ON mm/dd/yy AT 08:34   Page Size = 4K

     SPACE SUMMARY                             Percent         Reorg
                                  Number      of Total         Factor
                                    1            2               3
                               -----------     ---------       --------

    Pages with room ........          10   |       83.33   |       0
          with avg. room ...           0   |        0.00   |       1
          with small room ..           0   |        0.00   |       2
          with NO room .....           2   |       16.67   |       4

                 TOTAL .....          12   |      100.00   |     16.67

     UPDATE SUMMARY               Number     No. Updated      Pct Updated
                                   4             5               6
                               -------------   ----------      ---------
                   Pages...          12   |            0   |      0.0
                  Tracks...           1   |            0   |      0.0
              Cylinders...            1   |            0   |      0.0

      PF17 = Space Graph        PF22 = Space Map       ENTER = Space Graph
      PF18 = Update Graph       PF23 = Update Map
```

Elements of the table space summary report (continued)

1 Number of pages with room to insert a row of data of the stated size.

2 Percent of total pages with room to insert a row of data

3 Reorg factor for the space

4 Number of a given unit of space

5 Number of units updated since the last reorganization

6 Copy factor; the percent of the total updated since the last reorganization

Interpreting the table space summary report

Review these values when interpreting the table space summary report:

- The Pages with Room field

  This field assesses the likelihood DB2 will find room to insert a row in a page of the table space. It does not analyze the actual amount of free space available.

- The reorg factor

  For this example, the Pages with Room field shows that 10 pages in the table space can accept an inserted row of any length. The reorg factor for this tablespacege is 16.67. It indicates that 17% of the pages in this table space cannot accept an inserted row of any length. A large value for the reorg factor indicates that the table space cannot accept many inserts without adding more extents. You can find a detailed discussion of the reorg factor in the unit "Reorg Factor" on page 98.

- The copy factor

  The copy factor is an indicator of the degree of update of a table space. It is discussed in detail in the unit "Copy Factor" on page 100. The size of this value can indicate whether an incremental image copy is preferable to a full image copy. Unless the value is small and the table space large, the convenience of a full image copy probably outweighs the time savings realized by an incremental image copy. In this example, because the level of change is less than 1%, you may want to wait to perform an image copy or perform only an incremental copy.

Concepts about Graphs for Table Spaces

Overview

This unit provides general information about graphs for table spaces.

Background about the space graph for table spaces

Each graph is a single-panel representation of the whole dataset. In a single glance, you can see the overall picture of the free space or patterns for updates in a table space.

Purpose of the space graph for table spaces

Normally, the !DB/SMU graphs will not of themselves solve problems. However, you can use the graphs to answer these and other questios:

- What explains the free space pattern in the table space?
- Why are updates clustered in certain areas of the table space?
- Should 1 of the tables in the table space be separated to its own table space?
- When should a table space be reorganized?

Space Graph for Table Spaces

Overview

This unit discusses the space graph for table spaces

Prerequisite information

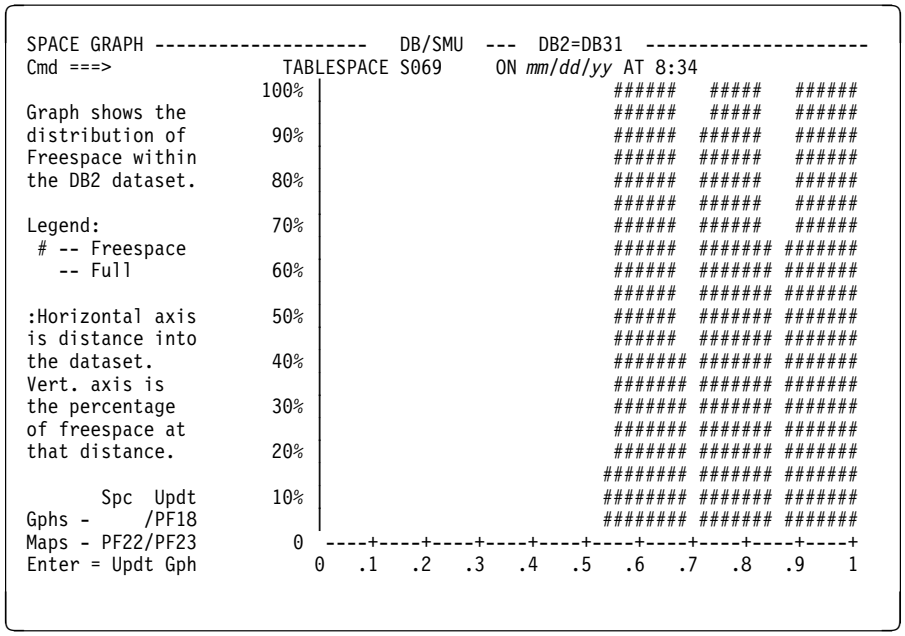
Before using this unit, read "Concepts about Graphs for Table Spaces" earlier in this chapter.

Accessing the space graph for table spaces

To access the space graph for table spaces, select option 2 on the Table Space Report Menu panel.

Elements of the space graph for table spaces

This is a space graph for table spaces.



Interpreting the space graph for table spaces

The graph consists of 50 columns, each representing 2% of the table space. For example, if the dataset is 200 cylinders, each column represents 4 cylinders (2% of 200).

Peaks in the graph show areas where there is abundant free space. Valleys show where there is a shortage of free space.

There can be no standard for deciding whether the free space in a given table space is adequate or not. The characteristics of the individual table space determine that. Normally, high update activity requires substantial free space evenly distributed over the dataset. The sample graph shows that the first 104 cylinders are full. The next cylinder is about 15% full. The balance of the remaining cylinders have abundant free space on almost every page.

Update Graph for Table Spaces

Overview

This unit discusses the update graph for table spaces

Prerequisite information

Before using this unit, read "Concepts about Graphs for Table Spaces" on page 278 earlier in this chapter.

Accessing the update graph for table spaces

To access the update graph for table spaces, select option 3 on the Table Space Report Menu panel.

Elements of the update graph for table spaces

This is an update graph for table spaces.

```
UPDATE GRAPH --------------------   DB/SMU  ---  DB2=DB31  ----------------------
Cmd ===>                TABLESPACE S069     ON mm/dd/yy AT 8:34
                      100% |
Graph shows the            |
distribution of        90% |
pages updated              |
since Image Copy       80% |

Legend:                70% |
 # -- Updated              |
   -- Unchanged         60% |

Horizontal axis        50% |
is distance into           |
the dataset.           40% |
Vertical axis is           |
the percentage         30% |
of pages updated           |#
at that distance       20% |#
                           |#
       Spc  Updt       10% |#
Gphs - PF17/               |#
Maps - PF22/PF23         0  ----+----+----+----+----+----+----+----+----+----+
Enter =Space Map           0   .1   .2   .3   .4   .5   .6   .7   .8   .9    1
```

Interpreting the update graph for table spaces

The graph consists of 50 columns, each representing 2% of the table space. For example, if the dataset is 200 cylinders, each column represents 4 cylinders (2% of 200).

Peaks in the graph show areas where there is high activity. Valleys show areas of the table space where there has been little update activity.

The sample graph shows that 25% of the first 4 cylinders has been updated. The balance of the space is unchanged.

Concepts about Maps for Table Spaces

Overview

There are 2 different kinds of maps for table spaces:

- Free space maps
- Update maps

These respectively show free space and update activity in the table space.

Using the zoom facility

The !DB/SMU zoom facility is available for viewing the usage map for table spaces. See “Zooming” on page 265 in the chapter “Using the Space Map Function” for information about the zoom facility.

Purpose of the the maps for table spaces

The maps for table spaces can assist you in answering these and other questions:

- Are pages full because of large VARCHAR columns?
- Are updates primarily on a single table in the table space?
- Should the table be in a separate table space?

Space Map for Table Spaces

Overview

This unit discusses the space map for table spaces.

Purpose of the space map for table spaces

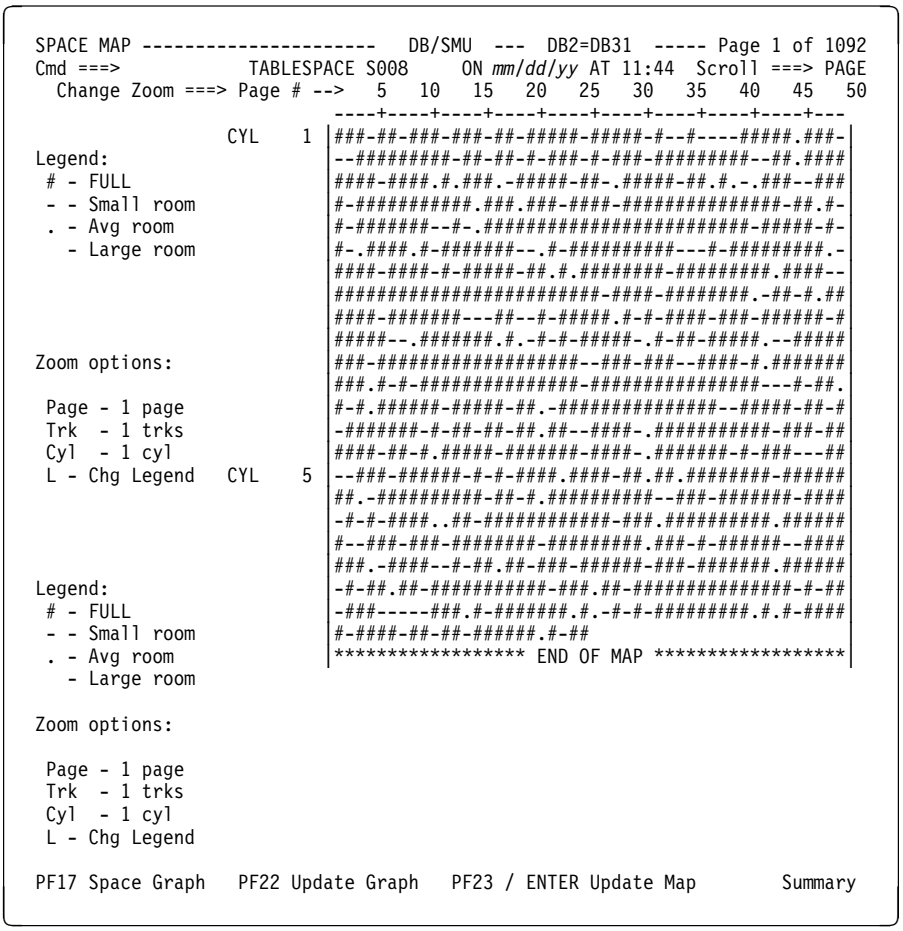
The space map provides a map of free space in a given table space.

Accessing the space map for table spaces

To access the space map, select option 4 on the Table Space Report Menu panel.

Sample map

This is a space map report for table spaces.



Interpreting the space map for table spaces

In the example shown above, page zoom is in effect. Each character in the map represents 1 page. There are 50 columns in the map. Since each cylinder consists of 150 pages, each cylinder is represented by 3 rows in the map. The beginning of a new cylinder is noted on the left side of the map (CYL 1, CYL 5, and so forth). The maximum number of pages is shown in the upper right corner.

The second line of the panel gives the table space name and the date and time of the analysis. The legend for this map is shown on the left side of the panel, and can be changed by selecting L as the zoom option. In this panel, "#" represents a full page, "." a page with average room, and "-" a page with very little room. Reviewing this map shows that this table space is quite full, with the majority of pages being full.

Update Map for Table Spaces

Overview

This unit discusses the update map for table spaces.

Purpose of the update map for table spaces

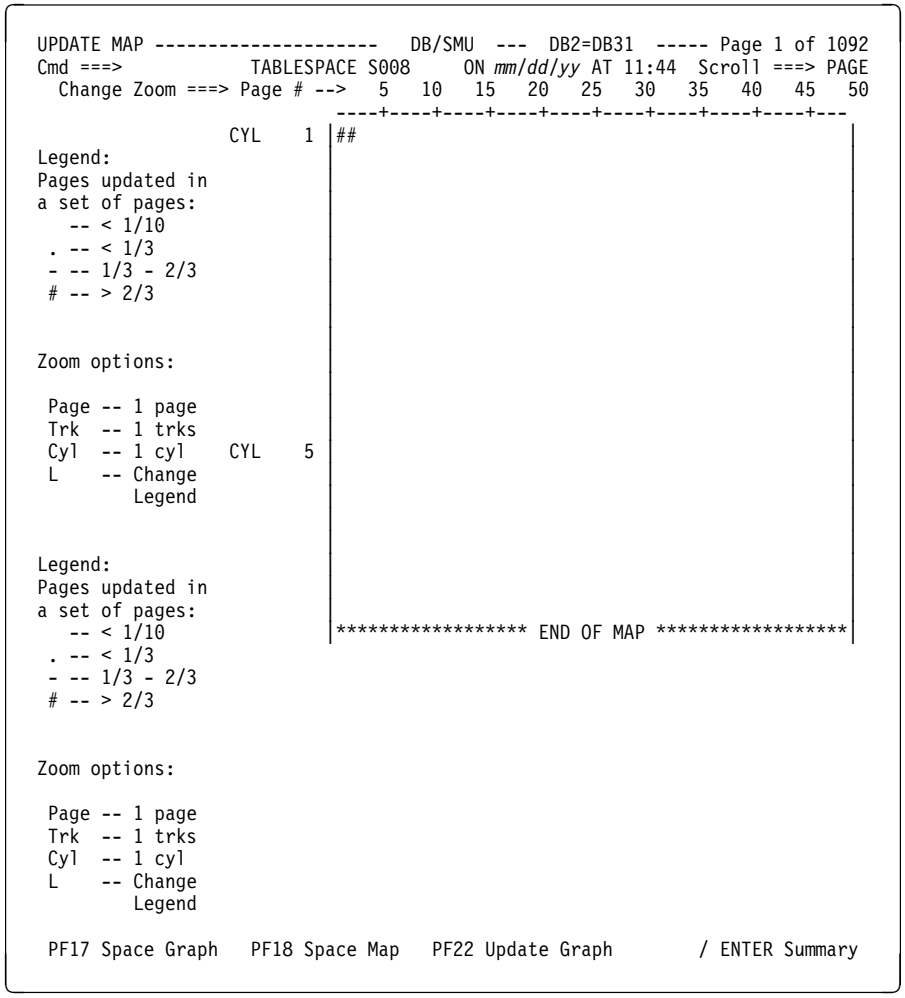
The update map provides a map of update activity in a given table space.

Accessing the update map for table spaces

To access the update map, select option 5 on the Table Space Report Menu panel.

Elements of the update map for table spaces

This is an update map for table spaces.



Interpreting the update map for table spaces

In this example, page zoom is in effect. Each character in the map represents 1 page. There are 50 columns in the map. Since each cylinder consists of 150 pages, each cylinder is represented by 3 rows in the map. The beginning of a new cylinder is noted on the left side of the map (CYL 1, CYL 5, and so forth). The maximum number of pages is shown in the upper right corner. The second line of the panel gives the table space name and the date and time of the analysis. The legend for this map is shown on the left side of the panel, and can be changed by selecting L as the zoom option. In this panel, "#" represents a full page and "." represents average room. Cylinder 1 has 2 pages with greater than 66% update. The balance of the pages have less than 10% update.

Chapter 16.
Space Map Analysis Reports for Index Spaces

Introduction

This chapter contains information about using the space map analysis reports for index spaces. It contains example reports, maps, and graphs.

Chapter Contents

Chapter Overview

Prerequisites for using this chapter

Before reviewing the information in this chapter, read the chapter "Using the Space Map Function" on page 261.

Generating index space space map analysis reports in batch

You can also use batch procedures to generate similar reports. For information on the batch versions, see "Using the Batch Reports for !DB/SMU" on page 591.

Organization of information in this chapter

The units in this chapter describe the index space reports generated by an online space map analysis. Each unit describes the way in which you access the report using !DB/SMU online interface and describes the contents of the report.

Types of space map analysis reports for index spaces

The index space space map analysis reports provide details to assist you in improving the performance of your DB2 data bases. Remember that, unlike table spaces, index spaces can contain only one index. This chart provides summary information about each space map analysis report and points you to detailed information.

Index Space Report	Report Contents	Detailed Information
Space Summary	Numerical summary of free space and updates	"Space Summary Report" on page 293
Usage Graph	Utilization of the DASD space allocated to the index dataset	"Usage Graph for Index Spaces" on page 295
Usage Map	Shows which pages of an index are used	"Usage Map for Index Spaces" on page 297

Using Space Map Reports for Index Spaces

Overview

Each time you run the !DB/SMU space map analysis function, it creates a series of index space reports. This unit describes how to access the reports and navigate through them.

Selecting the appropriate generation of reports

When you access the Report Select List panel, !DB/SMU presents you with a list of all available reports for an index space, sorted in date order with the most recent generation of reports at the top of the list. You can view these most recent reports or earlier ones.

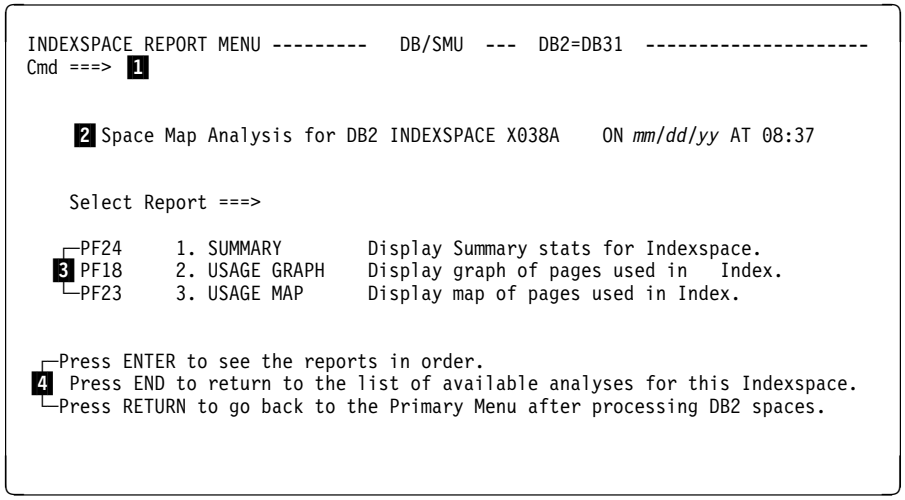
Accessing the space map analysis reports for index spaces

Follow these steps to access the space map analysis reports for index spaces.

Step	Action
1	Access the Report Select List panel as described in "Viewing a Report" on page 91.
2	In the list of available reports, type **S** in the SEL column next to the reports you want to see.
3	Press Enter. **Result:** !DB/SMU displays the Indexspace Report Menu panel.
4	On the Indexspace Report Menu panel, type the number of the report you want to see on the command line or press the appropriate PF key. If you want to view the reports in order, press Enter.

Elements of the Indexspace Report Menu panel

This is the Indexspace Report Menu panel.



1 Command line; type the number of the report you want to view here.

2 Name of index space for which the reports have been generated, and the date and time of this generation of reports

3 Basic set of available reports for index spaces

4 Additional actions you can perform on this panel. If you have selected multiple reports on the Report Select List panel, !DB/SMU displays all of the reports before executing the END or RETURN commands.

Space Summary Report

Overview

This unit describes the summary report for index spaces.

Purpose of the summary report for index spaces

The summary report for index spaces provides summary information about the use of space within the index space.

Accessing the summary report for index spaces

To access this report, select option 1 on the Indexspace Report Menu panel.

Elements of the index space summary report

This is the index space summary report.

```
INDEXSPACE SUMMARY -------------   DB/SMU  ---  DB2=DB31  ---------------------
Cmd ===>                                                            mm/dd/yy 08:37


                 INDEXSPACE X038A    ON mm/dd/yy AT 08:37:42
                 Index X038A              on table T038A

     USAGE SUMMARY
                                  Number          No. Used        Pct Used
                                     1               2               3
                             -----------       ----------       --------
               Pages...             108   |            93   |      86.11
              Tracks...               9   |             9   |     100.00
           Cylinders...               1   |             1   |     100.00


      This screen shows the number of        All other pages of the Index
      different Pages, Tracks, and           dataset are unused.  This
      Cylinders of the Index dataset         screen does not indicate how
      that have index data in them.          full any of these pages are.



    PF18 = Usage Graph       PF23 = Usage Map       ENTER = Usage Graph
```

1 Number of units in the space

2 Number of units currently in use

3 The percent of the total units used

Interpreting the index space summary report

The copy factor is a percentage of pages actually used in the index. It is discussed in detail in “Copy Factor” on page 100. In this example, the large values in the Pct Used field indicate that you should seriously consider reorganizing this index and allocating additional DASD to it.

Usage Graph for Index Spaces

Overview

This unit discusses the usage graph for index spaces

Purpose of the usage graph for index spaces

For an index space, the bits DB2 keeps indicate which pages are used; so the graph shows instantaneously the utilization of DASD space allocated to the index.

This !DB/SMU graph can help you answer these and other questions:

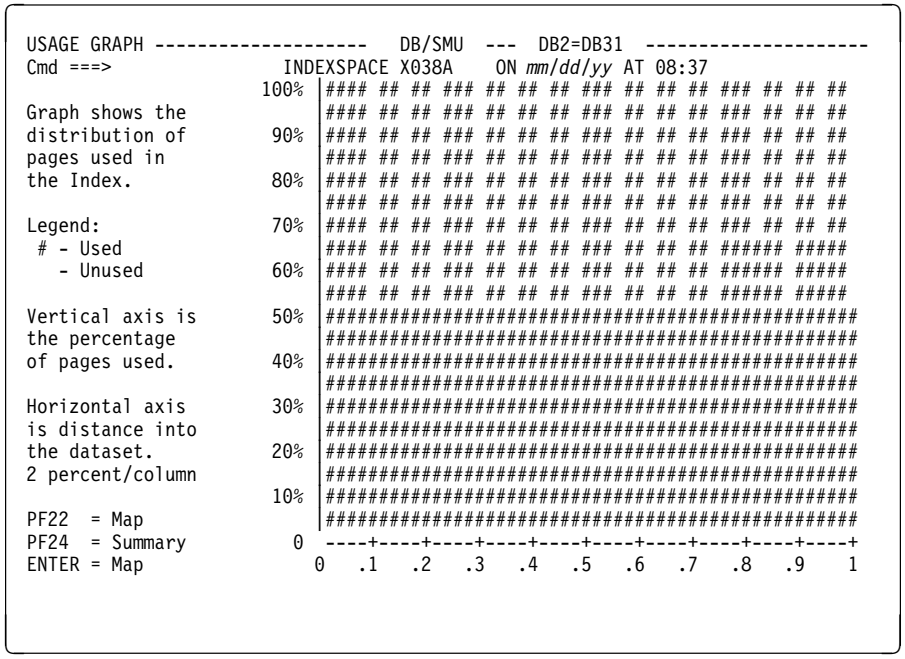
- Do the unused pages form a pattern?
- When should a index space be reorganized?

Accessing the usage graph for index spaces

To access the usage graph for index spaces, select option 2 on the Indexspace Report Menu panel.

Elements of the usage graph for index spaces

This is the usage graph for index spaces.



Interpreting the usage graph for index spaces

The graph consists of 50 columns, each representing 2% of the index space. For example, if the dataset is 200 cylinders, each column represents 4 cylinders (2% of 200). This graph shows the general distribution of pages in use in the index space. It shows immediately whether an index space is over-allocated, that is, more DASD has been allocated than is used.

Peaks in the graph show areas where there is high usage. Valleys show where there are few pages used.

The sample graph shows that, in this index space, many pages are used. This indicates good use of DASD space. That is, the space allocated is not excessive.

Additional pages remain to be used. If a graph is very full (many pages in use), it can indicate that the index may soon go into additional extents. Reallocation may be needed.

Usage Map for Index Spaces

Overview

This unit discusses the usage map for index spaces. The usage map shows which pages of an index are used.

Using the zoom facility

The !DB/SMU zoom facility is available for viewing the usage map for index spaces. See “Zooming” on page 265 in the chapter “Using the Space Map Function” for information about the zoom facility.

Purpose of the usage map for index spaces

The usage map provides a map of the pages in an index that are used. It permits you to answer the question, “Has too much space been allocated to the index?”

Accessing the usage map for index spaces

To access the usage map report, select option 3 on the Index Space Report Menu panel.

Elements of the usage map for index spaces

This is a usage map for index spaces

```
USAGE MAP ----------------------   DB/SMU  ---  DB2=DB31  ------ Page 1 of 108
Cmd ===>            INDEXSPACE X038A    ON mm/dd/yy AT 08:37  Scroll ===> PAGE
  Change Zoom ===> PAGE # -->   5   10   15   20   25   30   35   40   45   50
                            ----+----+----+----+----+----+----+----+----+----+---
                  CYL    1 |######## ###### ###### ###### ###### ###### ###|
Legend:                    |### ###### ###### ###### ###### ###### ###### ##|
Pages used in              |#### ######                                       |
a set of pages:            |****************** END OF MAP ******************|
   -- < 1/10
 . -- < 1/3
 - -- 1/3 - 2/3
 # -- > 2/3



Zoom options:

 Page -- 1 page
 Trk  -- 1 trks
 Cyl  -- 1 cyl
 L    -- Change
         Legend

        PF18 = Usage Graph     PF24 = Summary     ENTER = Summary
```

Interpreting the usage map for index spaces

The panel maps the index space usage (allocated pages). In the example shown above, page zoom is in effect. Each character in the map represents 1 page. There are 50 columns in the map. Since each cylinder consists of 150 pages, each cylinder is represented by 3 rows in the map. The beginning of a new cylinder is noted on the left side of the map (CYL 1). The second line of the panel gives the index space name and the date and time of the analysis. The legend for this map is shown on the left side of the panel, and can be changed by selecting L as the zoom option. In this panel, "#" represents a page in use. This example shows that cylinder 1 has many pages that are in use, but a number of pages are unused. If you reviewed a map with this pattern, you might determine that the index space could be smaller than curently allocated.

The Monitor Function

Chapter 17. Using the Monitor Functions

Introduction

This chapter contains information on using the monitor function, including the monitor exception reports, to identify exceptions and to manage DB2 utilities.

Chapter Contents

Chapter Overview

Prerequisites for using this chapter

Before running the monitor function, do the following:

- Ensure that you have appropriate authority for !DB/SMU
- Ensure that you have appropriate authority for dataset access required by a control system such as RACF

Organization of information in this chapter

This chapter describes the monitor function. It covers procedures for creating monitor exception reports and generating JCL to initiate selected DB2 utilities based on the results of a monitor run.

Online help and this chapter

This chapter contains descriptions of the online panels you use to view the monitor exception reports and the procedures you follow. Online help provides an additional level of detailed information about the panels and their fields.

Monitor functions in batch

When you run the batch utility MONITOR command, the monitor exception reports are produced in batch. Another byproduct of the batch MONITOR command is a DB2 table that contains summary values. See “Defining and Using Run Profiles for Batch Jobs” on page 491 and “Using the Batch Reports for !DB/SMU” on page 591 for information about the batch monitor function. See “Customized User-Written Reports” on page 631 to learn about ways to customize your reports using the DB2 table created by the monitor function.

Background for Using the Monitor Function

Overview

This unit describes ways in which you can use the monitor function.

Purpose of the monitor function

You can use the !DB/SMU monitor function to bring to your attention those exception conditions you identify and to manage many DB2 utilities. The output of using the monitor function consists of exception reports that identify DB2 table spaces and index spaces that satisfy certain criteria or exceed certain thresholds. Monitor exception reports provide exception information based on completed scans and space map analyses. To understand the use of scans and space maps, review the chapters "Using the Scan Function" on page 153 and "Using the Space Map Function" on page 261.

Process to access the monitor function

Here is the process to access the !DB/SMU monitor function.

Step	Action
1	On the !DB/SMU Primary Menu, select option 3 MONITOR.
2	When !DB/SMU displays it, complete the Monitor Selection panel as described in this chapter.
3	!DB/SMU displays the Monitor Exception Report listing the scans and space maps meeting the criteria you specified. Take appropriate actions as described later in this chapter.

Using the Monitor Selection Panel

Overview

This unit describes how to use the Monitor Selection panel to select a set of scan and space map reports you can view and act on.

Identifying selection criteria

You subset the reports you want to view by specifying selection criteria on the Monitor Selection panel. The values you specify on this panel (the exception selection criteria) are maintained as ISPF profile variables and retained from session to session.

Using pattern masks to select spaces

You can use !DB/SMU masking capabilities to create selection lists of spaces against which you will run the monitor function. Instead of specifying actual names in the fields of the Monitor Selection panel, you can specify pattern masks. The masking characters you use for the monitor function are the same ones you use for the scan and space map functions. For more information about masking, see “Using Masks to Create Selection Lists” on page 84.

Elements of the Monitor Selection panel

This is the Monitor Selection panel. The values you can provide for the exception selection criteria are listed following the panel and are keyed to it.

```
MONITOR SELECTION --------------   DB/SMU  ---  DB2=DSN2  -----------  -----
Cmd ===>

     This screen allows you to tailor the DB2 Dataset Exception Reports.
     Fill in any select data you want, then press ENTER for the report.
     Names can be masked using "*".  Name and range fields are optional.

         Dbdname ===> CJ23S*    Data Base name. 1
      Space name ===>           Tablespace or Indexspace name. 2
      Creator ID ===>           Creator Authorization name. 3

       Data Type ===> B         (T) Tablespaces   (I) Indexspaces  (B) Both. 4

          Source ===> B         (M) Space Maps    (S) Scans        (B) Both. 5

      Latest/All ===> L          6 Amount to Display: (L) Only the latest data.
                                                      (A) All recorded data.

 Ranges of values: 7                                 Format of
                                                     Dataset Size ===> P  8
    Reorg Factor ===> 0  to  ===> 100  (0-100)
     Copy Factor ===> 0  to  ===> 100  (0-100)       P - Pages     K - K bytes
  Number Extents ===> 3  to  ===> 255  (0-255)       T - Tracks    M - M bytes
                                                     C - Cylinders
```

Elements of the Monitor Selection panel (continued)

1 Enter a database name to restrict the list to spaces within a specific database.

2 Enter a table space or index space name to restrict the list to a single space.

3 Enter a creator ID to restrict the list to spaces with a specific creator ID.

4 Type **T** to include only table spaces, **I** to include only index spaces, or **B** for both.

5 Type **M** to use only space maps as source, or **S** to use only !DB/SMU scans as source, or **B** to use both space maps and !DB/SMU scans.

6 Type **L** to display only the most recent data; **A** to display all of the recorded data. !DB/SMU retains detail from both sources up to the value set for Max Reports in the system options.

Note: Using **A** can affect performance since it extracts *all* entries satisfying the selection criteria.

7 Select a minimum and a maximum value for each of the values shown:

Reorg factor
Enter a value from 0 through 100. The reorg factor does not apply to index spaces. See a discussion of the reorg factor in the unit "Reorg Factor" on page 98.

Copy factor
Enter a value from 0 through 100. See a discussion of the copy factor in the unit "Copy Factor" on page 100.

Number extents
Enter a value from 0 through 255. This is the number of DASD extents over all the volumes containing the DB2 space. See "Number of Extents" on page 102.

8 Select one of the following size codes:

P Pages
T Tracks
C Cylinders
K Kilobytes (1024 bytes)
M Millions of bytes

Using the Monitor Exception Report

Overview

This unit describes the ways in which you can use the monitor exception report to display reports, generate the JCL to perform an image copy or reorganization, or display or update utility profiles.

How the monitor function uses scan and space map reports

The monitor exception report lists the available scan and space map reports that meet the criteria you specified on the Monitor Selection panel. Each line of the report provides a summary of the performance characteristics of the underlying space. You can use this information to evaluate the fragmentation of the space, the need for an image copy, or the need for reorganization.

Elements of the monitor exception report

```
MONITOR EXCEPTION REPORT ---------  DB/SMU  ----  DB2=DSN2  -- REPORT 1  OF 11
Cmd ===>                                                      Scroll ===> CSR
 Selects:  S display detail reports; C/R generate DB2 Image Copy/Reorg.
 Commands: SORT(D) field; Locate field; SZ unit; UTIL OPTS profile; PRNT; DO.
                          Today is 90/11/01          (SZ)  (XT)   (CF)   (RF)
              T/                                   DATASET   NO.   COPY  REORG
 SEL  DBNAME  /I   NAME    PT TYPE   DATE   TIME     SIZE  XTNT FACTOR FACTOR
 1     2     3    4     5  6     7      8      9      10     11     12
 --- -------- -- -------- -- ---- -------- ----- ------- 13 === ------ ------
  _  CJ23SFC  TS DJTJUFTI    SCAN 99/10/20 11:07  13,110P   69     .0    2.0
  _  CJ23SFC  TS DBTTQBDF    SCAN 99/01/06 16:33  13,100P   67   20.0   82.5
  _  CJ23SFC  TS DBTTQBDF    MAP  99/10/22 17:05  13,070P   66   20.0   82.0
  _  CJ23SFC  TS CFIFNPUI    SCAN 99/01/10 15:05  13,600P   63     .0     .0
  _  CJ23SFC  IX CJHJOEYB    SCAN 99/09/30 00:53  13,450P   63   90.0     .0
  _  CJ23SFC  TS TFHTQD59    SCAN 99/06/15 03:27  12,006P   52   98.5   14.0
  _  CJ23SFC  TS SVOTUBUT    SCAN 99/07/06 18:09  13,440P   44     .0   23.0
  _  CJ23SFC  TS TFHTQD27    MAP  99/10/20 11:07  13,350P   33    3.0   14.5
  _  CJ23SFC  TS TFHTQD27    SCAN 99/10/20 11:00  13,350P   33    3.0   14.5
  _  CJ23SFC  TS OFXUFTU3    SCAN 99/10/20 11:07  13,290P   24     .0   12.0
  _  CJ23SKTB TS UFTUTQDF    SCAN 99/10/15 15:35  11,900P   17     .5   44.5
```

Elements of the monitor exception report (continued)

1 Type **S** in the Sel field next to a specific report to display the detailed !DB/SMU scan or space map for a given space. If detailed reports are not available, the dash (-) is omitted from the Sel field, indicating that you cannot see detailed reports for this space.

2 Names the database containing the DB2 space

3 Shows the type of DB2 space. The space may be:

TS Table space
IX Index space

4 Names the DB2 index space or table space

5 Shows the partition number, if any

6 Shows the source of the report—*scan* for a scan report or *map* for a space map report

7 Shows the date on which the report was created. You control the format of the date using a housekeeping option. See "Specifying !DB/SMU Configuration Options" on page 127.

8 Shows the time at which the report was created.

9 Shows the size of the entire space in the units defined in the panel. A code for the type of unit follows the size (pages, tracks, cylinders, K-bytes, or M-bytes).

10 Shows the number of extents associated with the table space or index space

11 Shows the copy factor for the space. The copy factor is explained in "Copy Factor" on page 100.

12 Shows the reorg factor of the space. The reorg factor is explained in "Reorg Factor" on page 98.

13 Double dashed line identifies the column on which the report is sorted; in this case NO. XTNT.

Using the Available Selects on a Monitor Exception Report

Overview

This unit describes the selects you can use in the SEL field of a monitor exception report.

Determining whether to use a select

If detailed reports exist for a DB2 table space or index space, a single dash (-) appears in the SEL field at the left of that spaces summary line. If the space in the SEL field is blank, you cannot use any selects against this space.

Determining what select to use

This table describes the selects you use to perform available tasks on the monitor exception report panel.

Task You Want To Perform	Action to Take
Display the selected scan or space map analysis reports	Type **S** in the Sel field for the report or reports you want. !DB/SMU continues processing by displaying the Table Space Stats or Index Statistis panel for scan reports, or the Tablespace Report Menu or Indexspace Report Menu for space map analyses.
For space map analyses only (identified as MAP on the panel), fast path to the space map reports	Type a number in the Sel field of the MAP report or reports you want. The value you type is one of the following: • Space map reports for table spaces 1. Summary 2. Space graph 3. Space map 4. Update graph 5. Update map • Space map reports for index spaces 1. Summary 2. Usage graph 3. Usage map !DB/SMU displays the appropriate space map report.

Determining what select to use (continued)

Task You Want To Perform	Action to Take
Generate the JCL for a full image copy of the selected DB2 table space or index space.	Type **C** in the Sel field of the report or reports you want. !DB/SMU displays the Utility Options panel so you can specify options for the utility.
Generate the JCL for an incremental image copy of the selected DB2 table space	Type one of the following in the Sel field of the report or reports you want. • I or P—!DB/SMU creates the JCL to perform an incremental image copy (only those pages that were modified since the previous image copy) • M—!DB/SMU creates the JCL to perform an incremental image copy *with accurate sizes*. (Since the amount of output produced by an image copy is unpredictable, selecting **M** causes !DB/SMU to calculate it accurately by reading the space map pages to determine the exact number that have been modified.)
Run the DB2 REORG utility against the selected table space	Type **R** in the Sel field of the table space report or reports you want.

Options for Running the DB2 Utilities from the Monitor Exception Report

Overview

This unit describes the steps you take to run a full or incremental image copy or a reorg from the Monitor Exception Report panel.

Using a select to initiate the utility

When you enter a select (C, I, P, M, or R) that initiates a DB2 utility, !DB/SMU displays the relevant utility option panel. !DB/SMU fills in the fields on the utility option panel with values that it takes from the *active utility profile.*

For information on setting the active utility profile from this panel, see the discussion on the UTIL OPTS command later in this chapter. "General Options for Creating DB2 Utility Profiles" on page 393 provides a complete discussion on creating, maintaining, and using DB2 utility profiles.

Process to initiate a utility

Follow these steps to execute a DB2 utility from the Monitor Exception Report panel.

Step	Action
1	Display the Monitor Exception Report panel.
2	(**This step is optional.**) On your first use of this function during the current session, or if you want to use a utility profile that is different from the current active profile, enter **UTIL OPTS** *profilename* on the command line. (If you omit the *profilename* parameter, !DB/SMU displays a selection list of profile names. If there is no list, !DB/SMU displays a profile definition panel for your use.)
3	Select a report to run a utility job stream against by typing one of the following select codes in the Sel field: • **C** for a full image copy • **I, P, or M** for an incremental image copy • **R** for the REORG utility
4	Press Enter. **Result**: !DB/SMU displays an options panel, for example, the Copy Utility Options, on which you can provide the values necessary to tailor the utility job.

Process to initiate a utility (continued)

Step	Action
5	Modify the options for the given DB2 utility and press Enter. **Result**: !DB/SMU continues to display options panels until you have completed all of the required options panels.
6	When the final options panel is complete, !DB/SMU stores the generated JCL to execute the utility job stream in the user PDS and displays the Process Member panel.
7	Enter an action on the Process Member panel to complete utility processing for this monitor exception report.

Using the Available Commands on a Monitor Exception Report

Overview

This unit describes the commands you can use from the monitor exception report.

Entering commands

To use any of the commands on the monitor exception report, type the command together with any valid operands on the command line of the monitor exception report. You can also initiate commands from the pop-up command panel. See "Using the DO command" on page 313 later in this chapter for information on using the pop-up menu.

Commands for the monitor exception report

The following commands can be typed in the command field of the monitor exception report. Use the DO command to display the command menu for this panel.

Task You Want to Perform	Command to Use	Additional Information
Display a pop-up menu of commands available on the monitor exception report panel.	DO	p.313
Find the first occurrence of a value equal to or greater than the value you specify	LOCATE	p.313
Print a report	PRNT	p.313
Reorder the information on the report	SORT/SORTD	p.314
Change the unit in which the size of a space is expressed	SZ	p.315
Set the active profile	UTIL OPTS	p.316

Using the DO command

Use the DO command to display the pop-up command menu for this panel. The command menu contains all of the command valid for use on the panel. To use the DO command menu, type the number of the command you want to execute on the command line of the pop-up menu.

Using the LOCATE Command

The LOCATE command (or its abbreviation, **L**) operates on the field that was the subject of the previous SORT. This field is identified by a line of double underscores (====). The default value for a LOCATE command is the Name field.

Entering the LOCATE command causes !DB/SMU to position the data on the panel so that the first line of data contains a value in the sort field that is greater than or equal to the specified LOCATE value. (If data on the panel is sorted in descending order, !DB/SMU positions to data so that the first line contains a value in the sort field that is less than or equal to the specified LOCATE value.)

Using LOCATE with decimal fractions

If you want to sort a panel using the copy factor or reorg factor (fields expressed with decimal fractions), use *only* the integer portion of the value as an argument on the LOCATE command. **L 22** is valid, but **L 22.3** is not.

Using the PRNT command

You can print the monitor exception report by typing the command **PRNT** on the command line of the monitor exception report panel. Note that using the PRNT command causes the *entire object list* to be printed, not just the portion of the list that you can view on a single panel.

Using the SORT/SORTD commands

Use the SORT command to sort the data in ascending order in any of the columns on the panel. Use the SORTD command to sort the data in descending order.

Using SORT/SORTD without an operand

You can use the SORT commands with or without an operand. When used without an operand, the commands sort on the NAME column of the report.

Using SORT/SORTD with an operand

To sort on a specific column of the monitor exception report, use the command in the form **SORT** *xx* where *xx* represents an operand from this chart.

Column You Want to Sort On	Operand to Use
Database name	DB
Type of space (table space or index space)	TI
Name of the table space or index space	NM
Partition number	PT
Source type (map or scan)	TY
Date of the report	DT
Time of the report	TM
Size of the dataset	SZ
Number of extents	XT
Copy factor	CF
Reorg factor	RF

Enhancing productivity using sorts

You can reorder the data on the monitor exception report in a number of different ways to ensure focus on those datasets needing attention. For example:

- Sort the data in descending order by the number of DASD extents to identify those table spaces or index spaces that need different space allocation or reorganization. Since datasets with many extents are almost always bad performers, you can quickly identify datasets that need work.
- Sort by reorg factor in descending order to identify table spaces that may need to be reorganized. For information on the reorg factor, see "Reorg Factor" on page 98.
- Sort by copy factor in descending order to see which table spaces need to be image copied. For information on the copy factor, see "Copy Factor" on page 100.

Using the SZ command

You can change the units used to express the size of spaces on the monitor exception report with the SZ command. The command takes the form **SZ *x***. Use these operands for the sizes you want to display.

Unit of Size You Want to Use	Operand to Use
Pages	P
Tracks	T
Cylinders	C
Kilobytes (1024 bytes)	K
Millions of bytes	M

Using the UTIL OPTS command

!DB/SMU can initiate DB2 utilities against table spaces and index spaces. !DB/SMU takes the options used to run the utilities from the active utility profile.

Setting the active utility profile

You can set the active utility profile in one of these two ways:

- Using the UTILITY option on !DB/SMU Housekeeping menu to display a profile
- Using the UTIL OPTS command on the monitor exception report to set the active utility profile

Format of the UTIL OPTS command

You can use the UTIL OPTS command from the command line of the monitor exception report in one of these two ways.

Command Format	Result
Type **UTIL OPTS** on the command line	!DB/SMU displays a selection list of existing utility profiles. You can select a profile from this list to be the active profile.
Type **UTIL OPTS** ***profilename*** on the command line, where *profilename* is the name of an existing profile in the system PDS	!DB/SMU sets the profile you named to be the active profile.

The Repair Function

Chapter 18. Fixing Error Pages

Introduction

This chapter contains information on using !DB/SMU to repair damaged DB2 pages.

Chapter Contents

Chapter Overview

Introduction

This chapter shows how !DB/SMU can help repair damaged DB2 pages quickly and efficiently. !DB/SMU allows you to make the repairs interactively. You can test changes for validity; if they are incorrect, you can clear, modify, or rework them. You can also fix multiple pages as a group. !DB/SMU generates the DB2 REPAIR utility input commands from the changes made in this process.

Advantages of using the !DB/SMU repair facility

This method enables you to fix the DB2 page much faster than if you printed a dump, analyzed the page by hand, and manually created complex, error-prone, control statements. The DB2 space is back online much faster and with much greater certainty of success.

As a precautionary measure, you can back up the table space or index space being repaired before actually making a repair. However, this is not strictly necessary. !DB/SMU generates the DB2 REPAIR utility JCL and control statements to effect the repair, and the REPAIR utility logs all changes. If you needed to back out the repair, you can use the DB2 RECOVERY utility to do so.

Overview of the data repair process

The !DB/SMU interactive repair facility works with an error list generated by the scan function. You can apply changes to the damaged table space or index space by using this error list, the online help, and the interactive repair panels. You can then test the fixes by using the CHK command. You continue using the facility in this way until you have corrected all the errors. The final output from !DB/SMU is the job stream for the DB2 REPAIR utility to make the changes you have specified. You can submit this job immediately, retain it for later submission, copy it, or simply discard it.

Note

You are not required to make changes. You can use the !DB/SMU repair function as an inquiry-only facility.

Online help for repairing data

You can use the repair facility even if you lack expert knowlege about the format and structure of DB2 data. The online help guides you in working on resolving the problem. Usually, a problem only requires a small change to the page in error, as shown in the first example in this chapter. The more experience you have or acquire in using !DB/SMU, the faster the solutions become obvious. Practicing making fixes, as described under “Interactive repairs of data integrity errors” on page 330, is highly recommended.

Correcting data integrity errors found in batch scans

You can also use the online repair facility to correct errors that were found in batch scans. (See “Introduction to Batch Operations for !DB/SMU” on page 455.) Begin the online correction process by selecting option 4, REPAIR, from the Primary Menu, and then select the table space or index space. You do not need to run a scan again online since you already have a list of the errors from the batch job.

Background on the examples for the repair facility

The repair facility is best described with examples. This chapter contains two complete examples of repair, both of which use the same actual case of a damaged DB2 page.

The first, simpler, example introduces the basic steps you always use to repair the page. This example describes the error list and the available online help panels. It also shows you how to use the page panels to test changes.

In the second example, a more complex approach to correcting the same error, shows the typical interactive nature of the repair process. It also shows the innate complexity of DB2 pages by presenting the steps that are required to delete a single DB2 row.

Simple Repairs (First Example)

Introduction

This unit describes the panels the system presents if it finds integrity check errors during a scan. It also contains an example of a simple repair of an error page.

Notification of data integrity errors

If !DB/SMU detects an error while scanning a table space or index space, it displays an error panel like one in the following example.

```
TABLE SPACE ERRORS -------------   DB/SMU  ---  DB2=D31A  --------------
Cmd ===>
    Data Base - KTS01DB

   Data Space - KTS01SUP  Partition  7

           on - 02/21/99


                         5,688 pages were scanned...

                             1 error pages were found...



                              Press ENTER to display errors.

                              Press END to display statistics.
```

Press Enter to display the list of errors.

Summary of data integrity errors

!DB/SMU summarizes the errors on the Data Spaces Errors panel, as shown in the following example. If there are many errors, you can use option 2 to identify those pages that have errors, or use option 3 to determine which errors have occurred.

```
DATA SPACE ERRORS -----------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===>                                                    Scroll ===> CSR

          Table Space  TDDB231A.DSNDBD.KTS01DB.KTS01SUP.I0001.A007

               DB/SMU detected integrity errors in this space.

  Select one of the following ===>                                      COUNT
                                                                        -----
                                    1. ALL      List All Errors.        3
                                    2. PAGES    List Pages With Errors. 1
                                    3. ERRORS   List Error Codes.       3

                                    4. BYPASS   Bypass Error Lists.
```

In this example, 1 page contains 3 errors. Select option 1 to diplay the All Errors panel, which lists all the errors found in the table space.

List of data integrity errors

The All Errors panel in the next figure contains a sample list of integrity check errors that were found during a scan. Only 1 page, number 60, contains the errors.

Example of the All Errors panel

The All Errors panel lists 3 errors. The first indicates a free space problem. The number preceding the message text corresponds to the message number in "!DB/SMU Data Integrity Checks" on page 249.

```
ALL ERRORS ------------------   DB/SMU  ---  DB2=DSN2  --------- ERROR 1 OF 3
Cmd ===>                                                    Scroll ===> CSR
  Selects:  S to see the error area;  ? for detail on the error.
  Commands: Locate page number.

SEL  PAGE      OFF  ERROR MESSAGE
     NUMBER    SET                HEX DUMP OF ERROR AREA
---  -------  ----  ------------------------------------------------------
 _   60             16  FREESPACE VALUE NOT = TOTAL FREESPACE
     (3C)     0000  10000180 4A0D0901 00003C02 00120F50 00004E00

 S   60             14  LARGE HOLE FOUND NOT ON HOLE CHAIN
     (3C)     0172  80003200 6608

 _   60             25  ORPHAN ID MAP ENTRY - NO RECORD FOR ENTRY
     (3C)     0FEE  0172

****************************** BOTTOM OF DATA  ******************************
```

Descriptions of data integrity errors

To display a description of an error, type **?** in the SEL field before the error message. The next examples show the descriptions for the sample errors.

```
-----------------------------   DB/SMU  ---  DB2=DSN2  ---------------------
Cmd ===>
----------------------  DB/SMU MESSAGES  --  Table Spaces ------------------

        14  LARGE HOLE FOUND NOT ON HOLE CHAIN

        Hole Error

        A large hole (record code X'80') was found in the table
        space, but it is not on the chain of holes anchored from the
        prefix in a two-byte field at +X'10'.

        There are two possibilities: either the record type flag
        could be damaged; or the hole chain anchor could be wrong.

        If error 16 (Freespace Value Incorrect) is also issued for
        this page, then the more likely error is that the hole
        record is the damage.

        The data displayed is the six bytes at the hole record prefix.
```

After you display the description of the error (in the example, error 14), you can also view the other error descriptions before proceeding to the Page Display panel on which you can make corrections.

```
-----------------------------   DB/SMU  ---  DB2=DSN2  ---------------------
Cmd ===>
----------------------  DB/SMU MESSAGES  --  Table Spaces ------------------

        16  FREESPACE VALUE NOT = TOTAL FREESPACE

        Freespace Error

        The freespace value in the header (a two-byte field at
        +X'0C') must match the accumulated value of the freespace in
        all the holes and the contiguous freespace area.  If the
        value is invalid, then DB/SMU shows message 16.

        The data displayed is the page header, the first 20 bytes of
        the page.
```

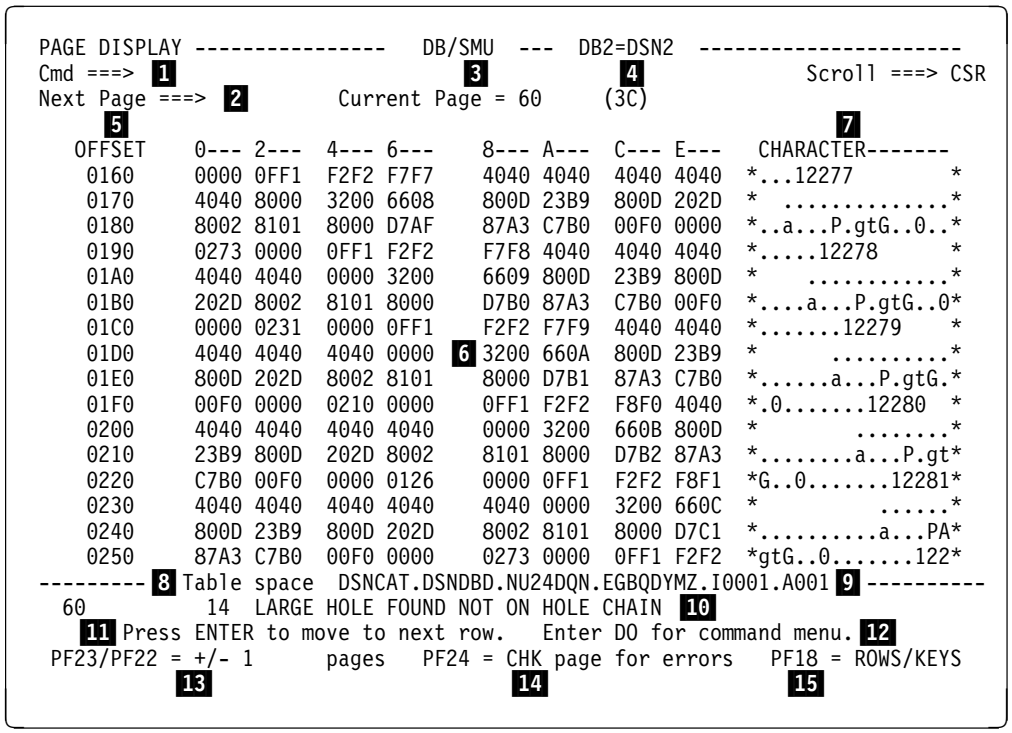
Help for data integrity errors

As the error descriptions indicate, the error reported by the second message could be the cause of the first. It appears there is an invalid hole record at offset 0172. To get a description of a table space page, use the help facility to display a list of the repair helps available.

There is apparently a free space problem and a hole error. (A hole is a record with a record ID of X'80'.) All such hole records should be chained from an anchor in the page header. The hole at 0172 is not chained from the hole anchor. Select the corresponding error entry and press Enter.

Page dump of pages with data integrity errors

!DB/SMU displays a dump of the error page when you use the **S** select to see the error area. For this example, !DB/SMU displays a dump of page 60 of the table space and places the cursor at the selected offset, in this case, 0172. The text of the error message selected on this panel is displayed near the bottom of this panel.



```
PAGE DISPLAY ----------------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===> 1                              3            4              Scroll ===> CSR
Next Page ===> 2        Current Page = 60     (3C)
     5                                                                 7
   OFFSET    0--- 2---  4--- 6---    8--- A---  C--- E---   CHARACTER-------
    0160     0000 0FF1  F2F2 F7F7    4040 4040  4040 4040  *...12277        *
    0170     4040 8000  3200 6608    800D 23B9  800D 202D  *  ..............*
    0180     8002 8101  8000 D7AF    87A3 C7B0  00F0 0000  *..a...P.gtG..0..*
    0190     0273 0000  0FF1 F2F2    F7F8 4040  4040 4040  *.....12278      *
    01A0     4040 4040  0000 3200    6609 800D  23B9 800D  *    ............*
    01B0     202D 8002  8101 8000    D7B0 87A3  C7B0 00F0  *....a...P.gtG..0*
    01C0     0000 0231  0000 0FF1    F2F2 F7F9  4040 4040  *.......12279    *
    01D0     4040 4040  4040 0000  6 3200 660A  800D 23B9  *       ..........*
    01E0     800D 202D  8002 8101    8000 D7B1  87A3 C7B0  *......a...P.gtG.*
    01F0     00F0 0000  0210 0000    0FF1 F2F2  F8F0 4040  *.0.......12280  *
    0200     4040 4040  4040 4040    0000 3200  660B 800D  *        ........*
    0210     23B9 800D  202D 8002    8101 8000  D7B2 87A3  *........a...P.gt*
    0220     C7B0 00F0  0000 0126    0000 0FF1  F2F2 F8F1  *G..0.......12281*
    0230     4040 4040  4040 4040    4040 0000  3200 660C  *           ......*
    0240     800D 23B9  800D 202D    8002 8101  8000 D7C1  *..........a...PA*
    0250     87A3 C7B0  00F0 0000    0273 0000  0FF1 F2F2  *gtG..0.......122*
--------- 8 Table space  DSNCAT.DSNDBD.NU24DQN.EGBQDYMZ.I0001.A001 9 ----------
  60          14  LARGE HOLE FOUND NOT ON HOLE CHAIN 10
   11 Press ENTER to move to next row.   Enter DO for command menu. 12
 PF23/PF22 = +/- 1      pages   PF24 = CHK page for errors   PF18 = ROWS/KEYS
           13                          14                          15
```

Online diagnostic information for data integrity errors

!DB/SMU saves you the inconvenience of referencing the DB2 *Diagnostic Guide* when you want to determine the layout of the DB2 pages. Access the help function on a Page Display panel to view the following help menu.

```
------------------------------   DB/SMU  ---  DB2=DSN2  ---------------------
Cmd ===>
------------------------------ PAGE REPAIR DISPLAY -------------------------


    The following are presented in sequence, or may be selected by number.


                        1.  Overview.
                        2.  Display Format.
                        3.  Positioning and Scrolling.
                        4.  Selecting a Page.
                        5.  Commands.
                        6.  Modifying Data.
                        7.  Table Spaces and Index Spaces.
                        8.  Table Space Page Formats.
                        9.  Index Space Page Formats.
                       10.  DB/SMU Checks.
                       11.  DB/SMU Error Messages.
```

Type the number of a menu item to get additional information on a topic. For example, menu selections 8 and 9 provide additional information on table space and index space formats.

Format for a table space page header

Expanded online help is provided with each repair panel. For example, if you use the repair facility to map and format a page header for a table space, online help is provided for the fomatted page header. A sample help panel describing the format of a table space page header is shown in the following two illustrations.

```
---------------------------------   DB/SMU  ---  DB2=D31A  ------------------
 CMD ===>
 ------------------- FORMAT OF TABLE SPACE HEADER PAGE HEADER ---------------
 PRESS END TO RETURN, ENTER TO CONTINUE

  OFFSET  LENGTH  NAME      DESCRIPTION
  ------  ------  --------  -------------------------------------------------
    00      1     PGCOMB    FLAGS  80 = I/O ERROR      10 = LAST BYTE OF
                                    40 = PAGE HAS BEEN       PAGE IS D5.
                                         REPAIRED       01 = PAGE WAS WRITTEN
                                                             BY DSN1COPY.
    01      6     PGLOGRBA  LOG RBA OF LAST PAGE UPDATE.
    07      1     PGLOGID   01 IF LOG RBA VALID, FF IF NOT (LOAD OR REORG).
    08      3     PGNUMBER  PAGE NUMBER OF THIS PAGE (IN HEX).
    0B      1     PGFLAGS   FLAGS  80 = BROKEN PAGE    20 = SEGMENTED SPACE
                                    18 = HEADER PAGE         HEADER PAGE.
    0C      2     HPGDBID   INTERNAL DATA BASE IDENTIFIER (DBID) OF DATA BASE.
    0E      2     HPGPSID   INTERNAL PAGE SPACE IDENTIFIER (PSID) OF TABLESPACE
    10      8               RESERVED.
    18      6     HPGTORBA  RBA TO WHICH RECOVERY HAS RECOVERED TABLE SPACE OR
                            PARTITION.
    1E      10    HPGTSTMP  TIMESTAMP AT WHICH THE DATABASE THAT HOLDS THIS
                            TABLE SPACE BECAME AN OWNER. IT IS
                            X'0001010100000000000' IF DATABASE IS NOT SHARED.
    28      4     HPGSSNM   DB2 SUBSYSTEM NAME.
    2C      2     HPGFOID   INTERNAL IDENTIFIER OF THE TABLE FILE DESCRIPTOR,
                            ALSO KNOWN AS OBID IN SYSIBM.SYSTABLSPACE.
    2E      2     HPGPGSZ   PAGE SIZE IN MULTIPLES OF 4KB. '=1' FOR 4K
                            '=8' FOR 32K PAGE SIZE.
    30      2     HPGSGSZ   THE NUMBER OF PAGES IN EACH SEGMENT OF SEGMENTED
                            TABLESPACE.  ZERO IF NOT SEGMENTED. ALSO IS SEGSIZE.
    32      2     HPGPARTN  NUMBER OF PARTITIONS IN A PARTITIONED TABLESPACE.
    34      3     HPGZPNUM  PAGE NUMBER OF FIRST DICTIONARY PAGE -0 IF NONE.
    37      1     HPGZPNUM  NUMBER OF 4K MEMORY PAGES TO STORE DICTIONARY.
    38      2     HPGTBLC   TABLE COUNTER. '=0' IF NO TABLE, '=1' IF EXACLY
                            ONE TABLE, AND '=2' IF MORE THAN ONE TABLE.
                            RESET TO '=0' IF ALL TABLES DROPPED'
    3A      2     HPGROID   INTERNAL IDENTIFIER OF TABLE IF HPGTBLC='1'.
    3C      4     HPGROID   RESERVED
    40      4     HPGMAXL   MAX RECORD LENGTH (NOT INCLUDING PREFIX)
                            POPULATED ONLY IF HPGTBLC='1'.
    42      2     HPGNUMCO  NUMBER OF COLUMNS IN THE TABLE IF HPGTBLC ='1
```

Format for a table space page header (continued)

```
---------------------------------   DB/SMU  ---  DB2=D31A  -----------------
Cmd ===>
------------------- FORMAT OF TABLE SPACE HEADER PAGE HEADER ----------------
Press END to return, ENTER to continue

 OFFSET  LENGTH  NAME      DESCRIPTION
 ------  ------  --------  ------------------------------------------------
   44      1     HPGFLAGS  Flage byte.
         1... ....  HPGSHARE  Share Indicator.
                             ON for shared tablespace.
                             OFF for non-shared tablespace.
         .1.. ....  HPGPSCON  Consistency indicator for data sharing.
                             ON means consistent tablespace which can be
                             opened in read-only mod by read only system.
                             OFF means it may be inconsistent tablespace
                             cannot be opened by read-only systems.
         ..1. ....  HPGDLVL   Down level indicator.
                             ON means tablespace may be down a level. Cannot
                             be opened by DB2 if HPGSHARE is ON. This flag is
                             set by DSN1COPY when the tablespace is restored
                             or page modified by REPAIR using a RESET function.
         ...1 ....  HPGEDITP  EDITPROC indicator. Applicable if HPGTBLC
                             ON means that there is an EDITPROC.
                             OFF means that there is not an EDITPROC.
         .... 1...  HPGUSERD  ON means that the dataset is User defined.
                             OFF means that the dataset is STROGROUP.
         .... 1...  HPGUIDX   Unique index indicator.
                             ON means there is a unique index.
                             OFF means that there is no unique index.
   45     *                   Reserved.
   48     10        HPGCONTM  Timestamp of setting consistency flag.
   52     8         HPGSGNAM  Storage Group name if it is a stogroup
                             defined dataset.
   5A     8         HPGVCATN  VSAM catalog name for dataset.
   62     6         HPGRBRBA  Recover base RBA updated by the buffer
   45     *                   manager.
```

You can use these help panels to familiarize yourself with the structure of DB2 pages. This will assist you in diagnosing the cause of the errors detected by !DB/SMU.

Approaches to page repair

There are two opposite approaches to fixing the damaged page in this example:

1. Assume that the hole is valid and fix all the free space values, for example, complete the delete.
2. Assume, more simply, that the hole is invalid and fix the hole record, for example, back out the delete.

Using the first approach involves fixing the total-free-space value and changing the ID map entry for the record. The ID map entry at offset FEE points to the hole record. (See "Example of the All Errors panel" on page 324 third error: code 25, offset 0FEE, data 0172.) This reinforces the conclusion that the hole that is invalid—ID map entries point to records, not to holes. Fixing the ID map entry would require several changes; this scenario is discussed under "Example of the All Errors panel" on page 324.

There is usually no *deductive* way to determine which of multiple possible causes of an integrity error is the *true* cause. In general, there is no right or wrong choice to make, rather only an easier or more difficult choice. Often knowledge of the applications that use the table space or index space in error can suggest which alternative approach is appropriate.

Interactive repairs of data integrity errors

Assume that it is the X'80' at offset 0172 that is incorrect. To change the X'80' at 0172 to X'00', simply type over the dump on the screen at offset 0172. After you make the change, the new value will be highlighted on the screen.

Using the CHK command to confirm data integrity corrections

Move the cursor to the command field and type **CHK,** or press the function key assigned to the CHK command, as shown on the bottom of the panel. If errors remain in the page, they are listed as they were on the All Errors panel. If there are no errors, the following message is highlighted in the upper, right corner of the Page Display panel: **`No Errors Found`**.

You can make prototype changes on any number of panel fields, on any number of pages. (See "Summary of navigating between pages" on page 347 for details on how to select a new page.) If you return to any page to which you made changes, your changes remain highlighted.

You can enter the CHK command as often as necessary and for any number of different pages. The CHK command does not perform cross-page checks. It checks only the single page that is displayed. This is especially relevant for index spaces.

After you complete the repair, ensure that all DB2 commits are done and the table space is stopped before you submit the job that implements the repair.

JCL for a simple repair

After all damaged pages are fixed and checked, press the End function key. !DB/SMU then generates JCL and statements to run the DB2 REPAIR utility. The job contents are based on the UTILITYR member in the !DB/SMU skels PDS and the changes you made on the Page Display panels. The following example contains sample statements for a repair job.

```
//JOBNAME JOB   'ACCOUNTING INFO',MSGLEVEL=(1,1)
//REPAIR  EXEC  PGM=DSNUTILB,PARM=DB2
//STEPLIB  DD   DISP=SHR,DSN=DSN.SSPGM
//KTSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSIN    DD   *

 REPAIR OBJECT LOG YES
    LOCATE TABLESPACE NU24DQN.EGBQDYMZ         PAGE  X'00003C'
        REPLACE  RESET
        VERIFY  OFFSET X'0172'  DATA X'8000'
        REPLACE OFFSET X'0172'  DATA X'0000'
```

You can invoke the ISPF editor to review the created JCL, and to edit, delete, print, log, or submit the job. The name of the member that contains the repair job stream is the current userid followed by "Z." !DB/SMU stores this member in the user PDS.

Safeguards within the Repair Facility

Overview

This unit reviews the fundamentals of the !DB/SMU repair facility.

Summary of safeguards for repairs

The preceding discussion should have allayed any concerns you might have about using the repair facility. Consider the following points.

1. Compared with DB2, !DB/SMU informs you of the existence of the error in a much less dramatic way (No application failed; no incorrect output was produced.) and of its nature in a more precise way. (There are almost 100 messages that precisely identify the location and nature of the error.)
2. Your prototype repair is made to a *copy* of the page diagnosed as in error. The real DB2 page is not changed unless DB2 security permits the repair, and you also generate and submit a job for the repair.
3. You are able to verify your prototype repair with an interactive repair capability (the CHK command), which informs you as to whether you are progressing towards fixing the error or creating new errors.
4. You can abandon the repair at any time, either by leaving the Page Display panel before completing the repair or by abandoning the repair job you generated.

Practicing Repairs

Introduction

This unit explains how you can simulate the repair of error pages to gain practice in making repairs.

Beginning a practice session for repairing data integrity errors

Even if there are no damaged pages, !DB/SMU allows you to simulate damaged pages and to practice repairing them. Run a scan. When the "OK" message displays in the Space Scan Complete panel, type **DUMP** in the command field to access the repair facility. (You can also select option 2. Repair from the Primary Menu to access the repair facility.)

```
SPACE SCAN COMPLETE --------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===>

    Data Base - DSNDB06
  Table Space - SYSPLAN            on -  mm/dd/yy


                          835 pages were scanned.

               * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
               * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
               * *                                                     * *
               * *  THE TABLE SPACE IS OK.  ALL TESTS PASSED.         * *
               * *          NO  ERRORS WERE DETECTED.                  * *
               * *                                                     * *
               * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
               * * * * * * * * * * * * * * * * * * * * * * * * * * * * *


                         Press ENTER to display stats.

                       Press END to return to the select list.
```

Navigating within the repair facility

!DB/SMU initially displays page 0 of the table space or index space. Type a page number in the Next Page field to display the page you want. In the Page Display panels shown in the following examples, the areas being discussed or modified are marked within the figure with "> <" and with "-->" preceding the line. On the actual panel, all modified fields are highlighted.

You can quickly locate any position in the page by typing **L** *ooo* (where *ooo* is the hexadecimal offset) in the command or Next Page field. You can place the cursor at the start of the next row in the page by pressing Enter.

Complex Repairs (Second Example)

Introduction

The following unit takes you through the process of making a complex repair.

Problem description of the complex repair example

As an example for a more complex error, we will fix the damaged page illustrated earlier in this chapter by deleting the damaged record at offset 0172.

This repair requires the following steps.

1. Add the length of the hole to the total free space.
2. Add the hole to the hole chain.
3. Free the ID map entry.
4. Add the free ID map entry to the free chain.

The first two changes must be made to the page header.

- The total free space value is a 2-byte value at offset 000C in the page. The current value is 0012.
- The hole chain anchor is a 2-byte value at offset 0010 in the page. The value is 0000. There is no hole chain; either (1) there have been no deletes from this page, or (2) the number of deletes has been at least matched by inserts of rows of the same length, or (3) the page was internally compressed by DB2.

The following example shows the page header before changes.

```
PAGE DISPLAY ----------------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===>                                                  Scroll ===> CSR
Next Page ===>        Current Page = 60     (3C)
   OFFSET    0--- 2---  4--- 6---    8--- A---  C--- E---   CHARACTER-------
--> 0000     0000 0180  0B56 F601    0000 3C02 >0012<0F50  *......6.........*
--> 0010    >0000<4E00  0000 3200    6601 800D  23B9 800D  *..+.............*
    0020     202D 8002  8101 8000    D7A8 87A3  C7B0 00F0  *....a...PygtG..0*
    0030     0000 0882  0000 0FF1    F2F2 F7F0  4040 4040  *...b...12270    *
    0040     4040 4040  4040 0000    3200 6602  800D 23B9  *      ..........*
```

The hole record at 00172 is 8000320066. The length of the hole is 0032 (a 2-byte value at offset 1 in the record). The offset in the page to the next hole is 0066 (a 2-byte value at offset 3 in the record).

Initial changes for the complex repair

The total free space value at offset 000C must be increased by 0032 to 0044 (such as, 0012 + 0032).

The hole chain anchor at offset 0010 must point to the hole, so the hole chain anchor is set to 0172, the offset of the hole record. The following example shows the page header after changes.

```
PAGE DISPLAY ----------------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===> CHK                                                 Scroll ===> CSR
Next Page ===>        Current Page = 60     (3C)
   OFFSET    0--- 2---  4--- 6---    8--- A---  C--- E---   CHARACTER-------
--> 0000     0000 0180  0B56 F601    0000 3C02 >0044<0F50  *......6.........*
--> 0010    >0172<4E00  0000 3200    6601 800D  23B9 800D  *..+.............*
    0020     202D 8002  8101 8000    D7A8 87A3  C7B0 00F0  *....a...PygtG..0*
    0030     0000 0882  0000 0FF1    F2F2 F7F0  4040 4040  *...b...12270    *
    0040     4040 4040  4040 0000    3200 6602  800D 23B9  *      ..........*
```

Checking the changes for the complex repair

Next, the changes are checked with the CHK command. The following example shows the list of errors produced from the CHK command after the changes were made.

```
ERRORS FOR PAGE 60 ----------   DB/SMU  ---  DB2=DSN2  --------- ERROR 1 OF 2
Cmd ===>                                                   Scroll ===> CSR
  Selects:  S to see the error area;  ? for detail on the error.
  Commands: Locate offset.

SEL  PAGE      OFF  ERROR MESSAGE
     NUMBER    SET                HEX DUMP OF ERROR AREA
---  -------  ----  ------------------------------------------------------
 _   60             15  ERROR IN HOLE CHAIN
     (3C)     0066  00000FF1 F2F2

 _   60             25  ORPHAN ID MAP ENTRY - NO RECORD FOR ENTRY
     (3C)     0FEE  0172
```

The list of errors indicates that a new error has been introduced. The hole chain is invalid at offset 0066.

Additional changes for the complex repair

Remember that the hole record at 00172 is 8000320066. The length of the hole is 0032; the offset in the page to the next hole is 0066. Here is the problem. Experience indicates that, since there is no valid record at offset 0066 in the page, the 0066 value is residue from the record that used to be at 0172, where the hole is now. We change the 66 to 00 at offset 0176 as shown in the next panel.

```
PAGE DISPLAY ----------------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===> CHK                                                Scroll ===> CSR
Next Page ===>        Current Page = 60     (3C)
   OFFSET    0--- 2---  4--- 6---    8--- A---  C--- E---   CHARACTER-------
    0160     0000 0FF1  F2F2 F7F7    4040 4040  4040 4040  *...12277        *
--> 0170     4040 8000  3200>0008<   800D 23B9  800D 202D  *  ..............*
    0180     8002 8101  8000 D7AF    87A3 C7B0  00F0 0000  *..a...P.gtG..0..*
    0190     0273 0000  0FF1 F2F2    F7F8 4040  4040 4040  *.....12278      *
    01A0     4040 4040  0000 3200    6609 800D  23B9 800D  *    ............*
    01B0     202D 8002  8101 8000    D7B0 87A3  C7B0 00F0  *....a...P.gtG..0*
    01C0     0000 0231  0000 0FF1    F2F2 F7F9  4040 4040  *.......12279    *
```

Confirming the corrections for the complex repair

Then, we can reissue the CHK command. The next example, shows the list of errors produced by the CHK command after making the last changes. The free space and hole chain errors have been cleared.

```
ERRORS FOR PAGE 60 ----------   DB/SMU  ---  DB2=DSN2  --------- ERROR 1 OF 1
Cmd ===>                                                  Scroll ===> CSR
  Selects:  S to see the error area;  ? for detail on the error.
  Commands: Locate offset.

SEL  PAGE      OFF  ERROR MESSAGE
     NUMBER    SET                HEX DUMP OF ERROR AREA
---  -------  ----  ------------------------------------------------------
 S   60             25  ORPHAN ID MAP ENTRY - NO RECORD FOR ENTRY
     (3C)     0FEE  0172

****************************** BOTTOM OF DATA  ****************************
```

Example repair for an ID map

Selecting the one remaining error from the error list would place the cursor at the now-faulty ID map entry. This field at offset 0FEE contains the value 0172, which is the offset to the record that the ID map entry owned. Note that the error list shows the value of the field at the offset shown.

To free an ID map entry, set it to 8000 or 0000. This will signify that this entry is at the end of the chain of free ID map entries. After this change is made, we again check the page with the CHK command.

```
PAGE DISPLAY ----------------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===> CHK                                                Scroll ===> CSR
Next Page ===>        Current Page = 60     (3C)
   OFFSET    0--- 2---  4--- 6---    8--- A---  C--- E---   CHARACTER-------
    0FD0     0460 042E  03FC 03CA    0398 0366  0334 0302  *.-.......q......*
--> 0FE0     02D0 029E  026C 023A    0208 01D6  01A4>8000< *...........O.u..*
    0FF0     0140 010E  00DC 00AA    0078 0046  0014 00C5  *. ............E*
    ********************* END OF PAGE DATA ***********************************
```

Completing the complex repair

The Errors for Page 60 panel shows the list of errors produced by the CHK command after the ID map entry was freed. The ID map entry is free, but the ID map free chain is invalid.

```
ERRORS FOR PAGE 60 ----------   DB/SMU  ---  DB2=DSN2  --------- ERROR 1 OF 1
Cmd ===>                                                    Scroll ===> CSR
  Selects:  S to see the error area;  ? for detail on the error.
  Commands: Locate offset.

SEL  PAGE      OFF  ERROR MESSAGE
     NUMBER    SET                HEX DUMP OF ERROR AREA
---  -------  ----  ------------------------------------------------
 S   60             26  ID MAP FREE CHAIN INVALID - FREE ENTRY NOT ON CHAIN
     (3C)     0FEE  8000

****************************** BOTTOM OF DATA  ******************************
```

Selecting the one remaining error from the error list would place the cursor at the freed ID map entry. The chain of free ID map entries starts at the 1-byte field at offset 0FFE. This value is 00. It should now contain the relative ID map number of the freed entry. The entry at offset 0FEE is the eithth entry, beginning at number 1 and counting back from the first entry at offset 0FFC.

Confirmation of the corrections for the complex repair

We change the value at 0FFE to 08, as shown below.

```
PAGE DISPLAY ----------------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===>                                                    Scroll ===> CSR
Next Page ===>        Current Page = 60     (3C)
   OFFSET    0--- 2---  4--- 6---    8--- A---  C--- E---   CHARACTER-------
    0FD0     0460 042E  03FC 03CA    0398 0366  0334 0302  *.-.......q......*
    0FE0     02D0 029E  026C 023A    0208 01D6  01A4>8000< *...........O.u..*
--> 0FF0     0140 010E  00DC 00AA    0078 0046  0014>08C5< *. ............E*
    ********************* END OF PAGE DATA ********************************
```

This time the CHK command finds the page to be valid.

JCL for a complex repair

Since all changes have been completed, we press the End function key. !DB/SMU then uses these changes to generate JCL and job statements for the DB2 REPAIR utility. !DB/SMU automatically invokes the ISPF editor so that you can review the job statements. From the ISPF panel, you can change, save, copy, or submit the job. After you submit a job, the job statements are saved in a member in the log PDS. The following example shows the contents of a sample member.

```
//JOBCARD  JOB  'ACCOUNTING INFO',MSGLEVEL=(1,1)
//REPAIR  EXEC  PGM=DSNUTILB,PARM=DB2
//STEPLIB  DD   DISP=SHR,DSN=DSN.SSPGM
//KTSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSIN    DD   *
 REPAIR OBJECT LOG YES
    LOCATE TABLESPACE NU24DQN.EGBQDYMZ         PAGE  X'00003C'
        REPLACE  RESET
        VERIFY  OFFSET X'000C'  DATA X'0012'
        REPLACE OFFSET X'000C'  DATA X'0044'
        VERIFY  OFFSET X'0010'  DATA X'0000'
        REPLACE OFFSET X'0010'  DATA X'0172'
        VERIFY  OFFSET X'0172'  DATA X'0000'
        REPLACE OFFSET X'0172'  DATA X'8000'
        VERIFY  OFFSET X'0176'  DATA X'6608'
        REPLACE OFFSET X'0176'  DATA X'0008'
        VERIFY  OFFSET X'0FEE'  DATA X'0172'
        REPLACE OFFSET X'0FEE'  DATA X'8000'
        VERIFY  OFFSET X'0FFE'  DATA X'00C5'
        REPLACE OFFSET X'0FFE'  DATA X'08C5'
```

Chapter 19. Using the Page Display Facility

Introduction

This chapter describes the capabilities of the !DB/SMU page display facility. Use this facility to display the various types of DB2 pages in order to browse or repair them.

Chapter Contents

Displaying DB2 Pages

Overview

This unit describes how to access and use the !DB/SMU page display facility.

Purpose of the page display facility

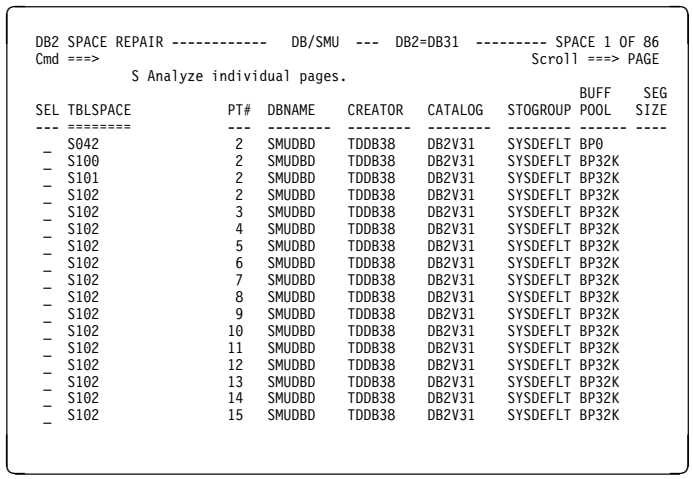
You can use the page display facility to assist in these major tasks:

- In support of the !DB/SMU repair function, you can examine a page to locate and correct a reported data integrity error.
- You can obtain information by browsing a table space or index space in native VSAM format or as formatted by !DB/SMU.

Accessing the page display facility

Follow these steps to access the page display facility.

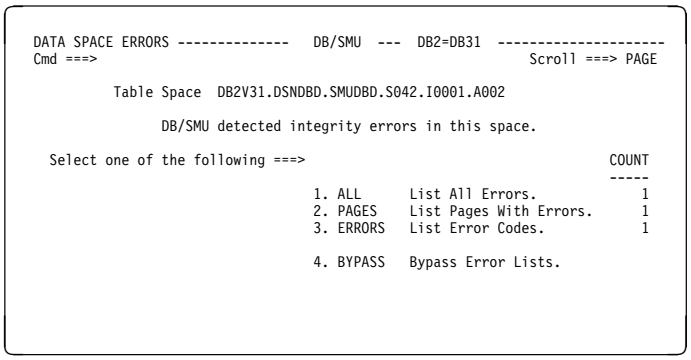
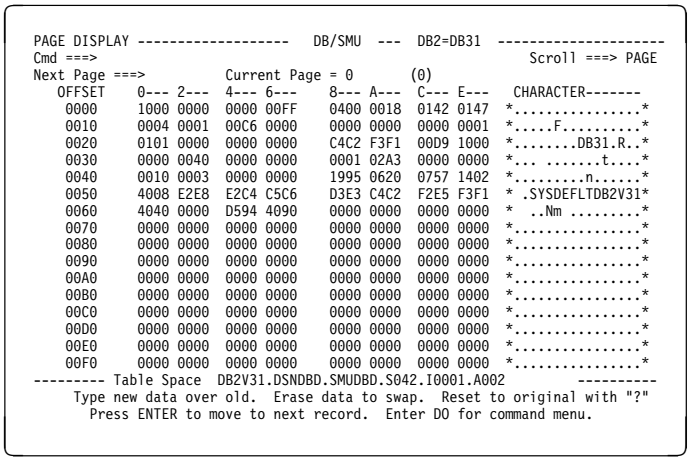
'

Step	Action
1	On the !DB/SMU Primary Menu, select option 4, REPAIR.
2	Press Enter. **Result:** !DB/SMU displays the Data Space Selection panel for the repair facility.
3	On the Data Space Selection panel, fill in the fields as appropriate.
4	Press Enter. **Result:** !DB/SMU displays the DB2 Space Repair panel. 

```
DB2 SPACE REPAIR ------------   DB/SMU  ---  DB2=DB31  --------- SPACE 1 OF 86
Cmd ===>                                                  Scroll ===> PAGE
          S Analyze individual pages.
                                                                    BUFF    SEG
SEL TBLSPACE            PT#  DBNAME    CREATOR   CATALOG   STOGROUP POOL   SIZE
--- ========            ---  --------  --------  --------  -------- ------ ----
 _  S042                 2   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP0
 _  S100                 2   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
 _  S101                 2   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
 _  S102                 2   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
 _  S102                 3   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
 _  S102                 4   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
 _  S102                 5   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
 _  S102                 6   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
 _  S102                 7   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
 _  S102                 8   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
 _  S102                 9   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
 _  S102                10   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
 _  S102                11   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
 _  S102                12   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
 _  S102                13   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
 _  S102                14   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
 _  S102                15   SMUDBD    TDDB38    DB2V31    SYSDEFLT BP32K
```

Accessing the page display facility (continued)

'

Step	Action
5	On the DB2 Space Repair panel, type **S** on the line next to the space you want to work with.
6	Press Enter. **Result:** !DB/SMU displays the Data Space Errors panel. 
7	On the Data Space Errors panel, select option 4, BYPASS.
8	Press Enter. **Result:** !DB/SMU displays the Page Display panel containing the page you want to work with. 

```
DATA SPACE ERRORS --------------   DB/SMU  ---  DB2=DB31  ----------------------
Cmd ===>                                                  Scroll ===> PAGE

          Table Space  DB2V31.DSNDBD.SMUDBD.S042.I0001.A002

               DB/SMU detected integrity errors in this space.

  Select one of the following ===>                                      COUNT
                                                                        -----
                                    1. ALL      List All Errors.            1
                                    2. PAGES    List Pages With Errors.     1
                                    3. ERRORS   List Error Codes.           1

                                    4. BYPASS   Bypass Error Lists.
```

```
PAGE DISPLAY -------------------   DB/SMU  ---  DB2=DB31  ----------------------
Cmd ===>                                                    Scroll ===> PAGE
Next Page ===>          Current Page = 0       (0)
   OFFSET    0--- 2---  4--- 6---    8--- A---  C--- E---   CHARACTER-------
    0000     1000 0000  0000 00FF    0400 0018  0142 0147  *................*
    0010     0004 0001  00C6 0000    0000 0000  0000 0001  *.....F.........*
    0020     0101 0000  0000 0000    C4C2 F3F1  00D9 1000  *........DB31.R..*
    0030     0000 0040  0000 0000    0001 02A3  0000 0000  *... .......t....*
    0040     0010 0003  0000 0000    1995 0620  0757 1402  *.........n......*
    0050     4008 E2E8  E2C4 C5C6    D3E3 C4C2  F2E5 F3F1  * .SYSDEFLTDB2V31*
    0060     4040 0000  D594 4090    0000 0000  0000 0000  *  ..Nm .........*
    0070     0000 0000  0000 0000    0000 0000  0000 0000  *................*
    0080     0000 0000  0000 0000    0000 0000  0000 0000  *................*
    0090     0000 0000  0000 0000    0000 0000  0000 0000  *................*
    00A0     0000 0000  0000 0000    0000 0000  0000 0000  *................*
    00B0     0000 0000  0000 0000    0000 0000  0000 0000  *................*
    00C0     0000 0000  0000 0000    0000 0000  0000 0000  *................*
    00D0     0000 0000  0000 0000    0000 0000  0000 0000  *................*
    00E0     0000 0000  0000 0000    0000 0000  0000 0000  *................*
    00F0     0000 0000  0000 0000    0000 0000  0000 0000  *................*
--------- Table Space  DB2V31.DSNDBD.SMUDBD.S042.I0001.A002         ----------
     Type new data over old.  Erase data to swap.  Reset to original with "?"
       Press ENTER to move to next record.  Enter DO for command menu.
```

Using the Page Display Facility

Overview

This unit describes the Page Display panel and provides information about navigating from page to page using the panel.

Elements of a Page Display panel

This is a sample Page Display panel.

```
PAGE DISPLAY ----------------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===> 1                          3            4              Scroll ===> CSR
Next Page ===> 2        Current Page = 60     (3C)
     5                                                          7
   OFFSET    0--- 2---  4--- 6---    8--- A---  C--- E---   CHARACTER-------
    0160     0000 0FF1  F2F2 F7F7    4040 4040  4040 4040  *...12277        *
    0170     4040 8000  3200 6608    800D 23B9  800D 202D  *  ..............*
    0180     8002 8101  8000 D7AF    87A3 C7B0  00F0 0000  *..a...P.gtG..0..*
    0190     0273 0000  0FF1 F2F2    F7F8 4040  4040 4040  *.....12278      *
    01A0     4040 4040  0000 3200    6609 800D  23B9 800D  *    ............*
    01B0     202D 8002  8101 8000    D7B0 87A3  C7B0 00F0  *....a...P.gtG..0*
    01C0     0000 0231  0000 0FF1    F2F2 F7F9  4040 4040  *.......12279    *
    01D0     4040 4040  4040 0000  6 3200 660A  800D 23B9  *      ..........*
    01E0     800D 202D  8002 8101    8000 D7B1  87A3 C7B0  *......a...P.gtG.*
    01F0     00F0 0000  0210 0000    0FF1 F2F2  F8F0 4040  *.0.......12280  *
    0200     4040 4040  4040 4040    0000 3200  660B 800D  *        ........*
    0210     23B9 800D  202D 8002    8101 8000  D7B2 87A3  *........a...P.gt*
    0220     C7B0 00F0  0000 0126    0000 0FF1  F2F2 F8F1  *G..0.......12281*
    0230     4040 4040  4040 4040    4040 0000  3200 660C  *          ......*
    0240     800D 23B9  800D 202D    8002 8101  8000 D7C1  *..........a...PA*
    0250     87A3 C7B0  00F0 0000    0273 0000  0FF1 F2F2  *gtG..0.......122*
--------- 8 Table space  DSNCAT.DSNDBD.NU24DQN.EGBQDYMZ.I0001.A001 9 ----------
  60          14  LARGE HOLE FOUND NOT ON HOLE CHAIN 10
   11 Press ENTER to move to next row.   Enter DO for command menu. 12
 PF23/PF22 = +/- 1      pages   PF24 = CHK page for errors   PF18 = ROWS/KEYS
           13                          14                      15
```

1 Location to enter ISPF and TSO commands and commands local to this panel. You can also enter most new page requests here.

2 Location to enter the number of the page you want to see next. This value can be specified as an absolute value (the actual page number you want) or a relative value (a page that is either +n pages or -n pages from the current page.

3 Current absolute page number in decimal form.

4 Current absolute page number in hexadecimal form.

Elements of a Page Display panel (continued)

5 Offset of this row from the beginning of the page.

6 A dump of the data page in hexadecimal format. Data in this area can be changed by typing directly over the existing data.

7 A character representation of the data shown in the hexadecimal portion of the page.

8 Name of the table space or index space you are working with.

9 The last four characters of the name provides, for partitioned spaces, the partition number you are working with, and for nonpartitioned multi-dataset spaces, the name of the *first* dataset in a multi-dataset page set.

10 In the example, an error message. This line can also contain the text "Type new data over old. Erase data to swap. Reset to original with ?". This information indicates that you can swap between the 2 most recent changes by typing **?** in one of the changed fields, or alter changed fields to their original values by using the Erase EOF key on your keyboard.

11 Pressing Enter moves your cursor to the hexadecimal value that begins the next row of the table. In order to use the Enter key, your cursor must be positioned somewhere within the hexadecimal portion of the page display.

12 Identifies the availability of a local command menu which is accessed by typing **DO** on the command line of this panel.

13 PF keys that act as toggle switches to move forward or backward one page.

14 PF key used to request that !DB/SMU check the current page for errors. Can be used when you have corrected errors by overtyping and want to ensure your correction is good.

15 PF key used to display the keys in an index space page or the rows in a table space page.

Background about page numbers

Before discussing ways to use page numbers to move between pages in a given DB2 space, you should understand these facts about them:

- Page numbers can be either absolute (a specific page number) or relative (a page number a known distance from the current page).
- When using relative positioning, and you want to move to a page number greater than the current page, prefix the relative page number with a plus sign (+). If the value for +nnn is too large, !DB/SMU displays an error message.
- When using relative positioning, and you want to move to a page number less than the current page, prefix the relative page number with a minus sign (-). If the value for -nnn is too large, !DB/SMU displays page 0.
- You can enter page numbers using either decimal or hexadecimal format.
- You can enter any page number, whether absolute or relative, in the Next Page field.
- You can also enter page numbers, except negative relative page numbers (for example, -225), on the command line. Entering values on the command line is especially convenient when used with the ISPF command delimiter. You set the ISPF command delimiter using ISPF option 0.1. The default value is the semicolon (;).
- To determine the size of a dataset, or to view the last page of a dataset, type LAST in the Next Page field. You *cannot* use LAST from the command line.
- If you have defined PF keys to perform the commands RIGHT and LEFT, you can use these keys to move forward and backward, respectively, in the DB2 space. Pressing the PF key results in a given amount of movement. That amount is initialized to 1 page at the start of every !DB/SMU session. You can change the value on the !DB/SMU panel by overtyping the value 1 (**13**) to any *decimal* number. You cannot use hexadecimal values. !DB/SMU retains the value you specify across DB2 spaces for the duration of the current session.

Summary of navigating between pages

This table summarizes the way you can navigate among DB2 pages in the table space or index space you are viewing. Before using this summary, be sure you understand the restrictions on the use of page numbers as described on the previous page.

Task You Want to Perform	Action to Take
Display a DB2 page other than the one you are currently viewing.	Perform one of the following: • Type the absolute or relative number of the page you want in the Next Page field; or • Type the absolute or relative number of the page you want on the command line (except negative relative page numbers or the value **LAST**).
Display a specific page.	Type the absolute page number in the Next Page field or on the command line.
Move 1 page forward in the space.	Perform one of the following: • Press the appropriate function key • Type +1 or +X'001' in the Next Page field or on the command line
Move 1 page backward in the space.	Perform one of the following: • Press the appropriate function key • Type -1 or -X'001' in the Next Page field.
Move to a page a known distance from the current page.	Perform one of the following: • Type +nnn or -nnn or type +X'nnn' or -X'nnn' in the Next Page field. • Type +nnn or +X'nnn' on the command line. (You cannot use a negative specification on the command line.)
Display the last page in the space.	Type **LAST** in the Next Page field.

Using the Commands Available from the Page Display Panel

Overview

This unit describes the tasks you can perform using the commands available from the Page Display panel.

Background about the page display commands

To assist you in browsing the contents of the Page Display panel, !DB/SMU offers a number of commands. Some are specific to this panel. Others, such as CANCEL or PRNT, are common to many !DB/SMU panels. Use the online help available with !DB/SMU to display a list of all of the commands available on the panel. To view the commands that are specific to this panel, type **DO** on the command line to view a pop-up list of "local" commands.

Tasks you can perform using !DB/SMU commands on the Page Display panel

To determine what tasks you can perform using !DB/SMU commands on this panel, review this table. The table points you to additional information on using each command.

Task You Want to Perform	Command to Use	Source of Additional Information
Discard modifications to a page display before they are processed	CANCEL	"Using the CANCEL Command" on page 350
Modify information in the hexadecimal portion of the page display	HEX	"Using the CHAR, HEX, and SWITCH Commands" on page 351
Modify information in the character portion of the page display	CHAR	"Using the CHAR, HEX, and SWITCH Commands" on page 351
Switch between hexadecimal and character portions of the page display	SWITCH	"Using the CHAR, HEX, and SWITCH Commands" on page 351

Tasks you can perform using !DB/SMU commands on the Page Display panel (continued)

Task You Want to Perform	Command to Use	Source of Additional Information
Search for a specified string	FIND	"Using the FIND Command" on page 352
Format the current display for a table space into its constituent rows	ROWS	"Using the ROWS Command" on page 372
Format the current display for an index space	XKEYS	"Using the XKEYS Command" on page 376
Position the cursor to a specific address on the DB2 page.	LOCATE	"Using the LOCATE Command" on page 354
Format header page data for a segmented or nonsegmented table space.	MAP	"Using the MAP Command Against Header Pages" on page 355
Format space map page data for a segmented or nonsegmented table space.	MAP	"Using the MAP Command Against Space Map Pages" on page 359
Format data page data for a segmented or nonsegmented table space.	MAP	"Using the MAP Command Against Data Pages" on page 365
Print the current page	PRNT	"Using the PRNT Command" on page 369
Discard any modifications made to the currently displayed page during this session	RESET	"Using the RESET Command" on page 371

Using the CANCEL Command

Overview

This unit provides information on using the CANCEL command when you are using a Page Display panel.

Purpose of the command

Use the CANCEL command if you do not want !DB/SMU to process your current modifications to a Page Display panel. The CANCEL command discards all modifications made to the currently displayed page since the most recent interrupt (for example, pressing ENTER or a function key). !DB/SMU saves any modifications made prior to the interrupt.

Format of the command

The CANCEL command can be abbreviated as CAN.

Issuing the command

Follow this step to issue the command.

Step	Action
1	Type **CANCEL** or **CAN** on the command line of the Page Display panel. ***Note:*** CANCEL cannot be issued from the DO command menu.

Contrasting CANCEL with RESET

In contrast to the CANCEL command, the RESET command discards any modifications to the current page any time during a given !DB/SMU session regardless of intervening interrupts. For information, see “Using the RESET Command” on page 371.

Using the CHAR, HEX, and SWITCH Commands

Overview

This unit provides information on using the CHAR, HEX, and SWITCH commands when you are using a Page Display panel.

Purpose of the commands

You can make prototype modifications to a DB2 page simply by overtyping values on the Page Display panel. Generally, You can modify values in the hexadecimal portion of the display, but character values are protected. However, you can make character values modifiable and move between portions of the page display using the CHAR, HEX, and SWITCH commands.

- Issuing CHAR places you in character mode, in which you can make modifications to the character section at the right side of the panel. CHAR positions your cursor at the first position of the character section.
- Issuing HEX places you in hexadecimal mode, in which you can make modifications to the hexadecimal section in the center of the panel. Hexadecimal mode is the default. HEX positions your cursor at the first position of the hexadecimal section.
- SWITCH switches between hexadecimal and character modes and moves your cursor to the appropriate section of the panel.

Format of the commands

The CHAR, HEX, and SWITCH commands cannot be abbreviated.

Issuing the commands

Follow this step to issue the commands.

Step	Action
1	Perform one of the following: • Type **CHAR**, **HEX**, or **SWITCH** on the command line of the Page Display panel; or • Type **DO** on the command line of the Page Display panel. Then type the number of the command you want in the Select field of the pop-up menu.

Using the FIND Command

Overview

This unit provides information on using the FIND command when you are using a Page Display panel.

Purpose of the command

The FIND command searches the current DB2 page for a specified string.

Format of the command

The FIND command can be abbreviated as F. Its syntax is similar to that of the FIND command used with ISPF browse or edit. The command takes the form FIND followed by the text string you want to find and optionally includes a parameter that can either precede or follow the search string.

Format of the search string

- You can optionally enclose the search string within delimiters (apostrophes or quotes).
- You can optionally specify the *type* of string as shown in this table. If you do not explicitly specify the string type, then its type is determined by default as described later in this unit.

Type of String	Explicit specification
Text Case is not significant for strings specified as text.	T'...'
Character Case *is* significant for strings specified as character.	C'...'
Hexadecimal Case is not significant for strings specified as hexadecimal.	X'...'

Default string type for the search string

On a Page Display panel the default string type for FIND is *hexadecimal*, not text. However, if the string has no delimiters, it is assumed to be hexadecimal if it contains only hexadecimal values. Otherwise, it is treated as text.

Optional parameters

Use the optional parameters described in this table with the FIND command. An optional parameter can either precede or follow the search string. If no optional parameter is specified, the default, NEXT, is assumed.

Optional Parameter	Meaning
NEXT	Find the next (following the current cursor position) occurrence of the search string on the page.
PREV	Find the previous (preceding the current cursor position) occurrence of the search string on the page.
FIRST	Find the first occurrence of the search string on the page.
LAST	Find the last occurrence of the search string on the page.
ALL	Find all occurrences of the search string on the page.

Issuing the command

Follow this step to issue the command.

Step	Action
1	Perform one of the following: • Type **FIND** *string* on the command line of the Page Display panel; or • Type **DO** on the command line of the Page Display panel. Then type the number of the command you want in the Select field of the pop-up menu.

To repeat the FIND command with its current search string and parameters, press the RFIND function key (normally F5 or F17) if you have assigned one.

Using the LOCATE Command

Overview

This unit provides information on using the LOCATE command when you are using a Page Display panel.

Purpose of the command

Use the LOCATE command to address any position on the current DB2 page.

Format of the command

The LOCATE command can be abbreviated as L. The syntax of the command is **L** *xxx* where *xxx* is any valid hexadecimal offset. For example, **L 4f3** positions the panel at the line prior to 4F0 and places the cursor under position 4F3.

Issuing the command

Follow this step to issue the command.

Step	Action
1	Perform one of the following: • Type **LOCATE** *nnn* or **L** *nnn* either on the command line or in the Next Page field of the Page Display panel; or • Type **DO** on the command line of the Page Display panel. Then type the number of the command you want in the Select field of the pop-up menu.

Using the MAP Command Against Header Pages

Overview

This unit provides information on using the MAP command against header pages on a Page Display panel.

Restriction on the use of the MAP command

The MAP command applies only to table spaces.

Purpose of the command

The output from the MAP command can help resolve data integrity errors that !DB/SMU finds and identifies in an error message. The MAP command formats the data in a space map page for a segmented or nonsegmented table space. There is a different type of output for the:

- Header page
- Space map pages
- Data pages

Information about mapping space map pages and data pages are found in the following units.

Format of the command

The MAP command cannot be abbreviated.

Prerequisite for issuing the command

To issue the MAP command for a header page, you must first display a header page on the Page Display panel. Page 0 of every DB2 table space is a header page.

Issuing the command

Once you have displayed a header page using the Page Display panel, follow this step to issue the command.

Step	Action
1	Perform one of the following: • Type **MAP** on the command line of the Page Display panel; or • Type **DO** on the command line of the Page Display panel. Then type the number of the command you want in the Select field of the pop-up menu.

Example of a mapped DB2 header page

The output of the MAP command against a header page results in a mapped Header Page panel. An example is shown here.

```
HEADER PAGE -----------------   DB/SMU  ---  DB2=DSN2  ----------- LINE 1 OF 3
Cmd ===>                                                    Scroll ===> CSR
 Page = 0         (0)          Selects: S Set position; F Format an item.
PF23/PF22 = +/- 1      pages.  Cmds: EXPAND/RESET All/Summary data; PRNT, DO.
    OFF
SEL SET         ITEM        HEX                               CHAR
--- ---- ------------------ --------------------------------  ----------------
 F  0000 ** PAGE HEADER **  0000436F2A4A500100000018017E0014 *...?..&......=..*
 _  001E ** RESERVED **     00000000000000000000000000000000 *................*
 _  0FFF ** PAGE TRAILER ** C5                                *E*
 ********************** BOTTOM OF HEADER PAGE DATA ************************
```

Selects available on a mapped header page

These selects are available on a mapped header page.

Select	Result of Using this Select
F	Selecting a line on the mapped Header Page panel causes !DB/SMU to decode the selected line into its constituent fields. An example Page Header panel is shown below.
S	!DB/SMU returns to the Page Display panel with the cursor positioned at the offset indicated by the mapped Header Page panel.

Example of the format of a page header

This is an example Page Header panel. You can add column type and attributes (length-, null-, and default-specification) values to the panel by typing **EXPAND** on the command line. You can remove these values by typing **RESET** on the command line.

```
PAGE HEADER -----------------   DB/SMU  ---  DB2=D31A  --------- FIELD
Cmd ===>                                                  Scroll ===> CSR
 PAGE  ITEM
   OFFSET    DATA                  DESCRIPTION
 ----------  --------------------  ------------------------------------
 0000  0000  00                    Flags
 0001  0001  000062C8A9E9          Log RBA
 0007  0007  01                    Log RBA Validity: Valid
 0008  0008  000000                Page Number
 000B  000B  1A                    Flags
                  18                    Header Page
                  02                    Page Has Been Modified
 000C  000C  010B                  DBID
 000E  000E  0018                  PSID
 0010  0010  00000001              High Preformatted Page
 0014  0014  00                    Reserved
 0015  0015  C6                    Release Mark
 0016  0016  0000                  Reserved
 0018  0018  000000000000          RBA Used by Recovery
 001E  001E  0001010100000000000   Timestamp DB Became ROSHARE
 0028  0028  C4F3F1C1              DB2 Subsystem Name
 002C  002C  0017                  Internal ID of Table File Desc
 002E  002E  1000                  Page Size 4K
 0030  0030  0000                  Nbr Pages in Segment
 0032  0032  0000                  Number of Partitions
 0034  0034  000000                Page Nbr of 1st Dictionary Page
 0037  0037  00                    Nbr Pages for Dictionary
 0038  0038  0002                  Table Counter
 003A  003A  0028                  Internal ID of Table
 003C  003C  00000000              Reserved
 0040  0040  0145                  Maximum Record Length
 0042  0042  004F                  Nbr of Columns in Table
 0044  0044  00                    Flag Byte
 0045  0045  000000                Reserved
 0048  0048  19990126074128912743  Timestamp Consistency Flag Set
 0052  0052  D2E3E2F0F1E2C7E3      STOGROUP Name
 005A  005A  E3C4C4C2F2F3F1C1      VSAM Catalog Name
 0062  0062  000062C926AF          Recover RBA Set by Buffer Mgr
 ****************************** BOTTOM OF DATA ************************
```

Commands available on the panels associated with the MAP command

This table describes the commands available to you when using the MAP command against a header space.

Command	Available on Panel	Description
EXPAND	Header Page panel	Entering EXPAND on the command line adds column type and attribute values to the panel.
PRNT	All panels described in this unit	Entering PRNT on the command line causes the entire contents of the page, not just that portion currently displayed, to be immediately sent to a JES printer. See "Using the PRNT Command" on page 369.
RESET	Header Page panel	Entering RESET on the command line returns the panel to displaying only summary information.

Using the MAP Command Against Space Map Pages

Overview

This unit provides information on using the MAP command against space map pages on a Page Display panel.

Restriction on the use of the MAP command

The MAP command applies only to table spaces.

Purpose of the command

The output from the MAP command can help resolve data integrity errors that !DB/SMU finds and identifies in an error message. The MAP command formats the data in a space map page for a segmented or nonsegmented table space. There is a different type of output for the:

- Header page
- Space map pages
- Data pages

Information about mapping header pages is found in the preceding unit; information about mapping data pages is found in the following unit.

Format of the command

The MAP command cannot be abbreviated.

Prerequisite to issuing the command

To issue the MAP command for a space map page, you must first display a space map page on the Page Display panel. Page 1 of every DB2 table space is a space map page.

Issuing the command

Once you have displayed a space map page using the Page Display panel, follow this step to issue the command.

Step	Action
1	Perform one of the following: • Type **MAP** on the command line of the Page Display panel; or • Type **DO** on the command line of the Page Display panel. Then type the number of the command you want in the Select field of the pop-up menu.

Example mapped space map page for a nonsegmented table space

The output of the MAP command issued against a space map page for a *nonsegmented* table space is a Space Map Page panel.

```
SPACE MAP PAGE ---------------   DB/SMU  ---  DB2=DSN2  ------ PAGE 1 OF 1,210
Cmd ===>                                                      Scroll ===> CSR
Page = 1         (1)       Pages Covered = 10760  Pages Used = 1210  1
Commands: Display next (NEXT) or prior (PREV) space map page.
Selects:  Display page (P).  Set position in space map (S) or update map (U).
                         PAGE           SPACE        UPDATE
               SEL  DECIMAL  HEX    OFFSET VALUE  OFFSET VALUE
                                       2            3
               ---  --------------  ------------  ------------
                _          0 000000  001A.0   3    0A9C.0   1 ─┐
                _          1 000001  001A.2   3    0A9C.1   1  │
                _          2 000002  001A.4   3    0A9C.2   1  │
                _          3 000003  001A.6   3    0A9C.3   1  │
                _          4 000004  001B.0   3    0A9C.4   0  │
                _          5 000005  001B.2   3    0A9C.5   1  │
                _          6 000006  001B.4   2    0A9C.6   1  │
                _          7 000007  001B.6   3    0A9C.7   1  4
                _          8 000008  001C.0   3    0A9D.0   0  │
                _          9 000009  001C.2   3    0A9D.1   0  │
                _         10 00000A  001C.4   0    0A9D.2   1  │
                _         11 00000B  001C.6   3    0A9D.3   1  │
                _         12 00000C  001D.0   3    0A9D.4   1  │
                _         13 00000D  001D.2   1    0A9D.5   0  │
                _         14 00000E  001D.4   3    0A9D.6   1  │
                _         15 00000F  001D.6   3    0A9D.7   1 ─┘
```

Elements of a mapped space map page for a nonsegmented table space

1 Header showing the Pages Covered for this space map page (usually 10,760 pages) and the Pages Used (1210)

2 Offset in the space map page of the 2 space management bits for the page; the value column shows the degree of available space in the page.

3 Offset of the update bit for the page. A 0 in the value column indicates the page has not been updated; a 1 indicates it has.

4 One line for each page controlled by the space map page.

Selects for a mapped space map page for a nonsegmented table space

Select	Description
P	Display the subject page.
S	Return to the Page Display panel for the space map page and place the cursor at the space bits.
U	Return to the Page Display panel for the space map page and place the cursor at the update bits.

Commands for a mapped space map page for a nonsegmented table space

Command	Description
NEXT	Change the subject of the Space Map Page panel to the next space map page.
PREV	Change the subject of the Space Map Page panel to the previous space map page.

Using function keys for the NEXT and PREV commands

If you have assigned function keys to the RIGHT and LEFT commands, you can use these keys to execute the NEXT and PREV commands respectively.

Example mapped space map page for a segmented table space

The output of the MAP command issued against a space map page for a *segmented* table space is a Space Map Page panel. This is a partial example.

```
SPACE MAP PAGE ---------------   DB/SMU  ---  DB2=DSN2  ----- SEGMENT 1 OF 338
Cmd ===>    .                                                    Scroll ===> CSR
 Commands: NEXT/PREV space map page; OBID obid or RESet; Locate offs; PRNT; DO.
 Selects:  Display page (P).  Set position in space map (S) or update map (U).

 Page=1           (1)       Segment Size = 8          Segments Covered = 338    1
 Free Segments=292    First Free Segment = 00002F  Last Formatted Page = 000143
            FIRST            NEXT
SEL OFFSET   PAGE SEGMENT  SEGMENT  OBID FLAGS SPACE MANAGEMENT/UPDATE DATA
--- ------ ------ -------- -------- ---- ----- +---+---+---+---+---+---+---+---
 _   001C  000002 00000001 00000002 0096  C0   FFFFFFFF   ┐
                                               11110000   │
 _   0027  00000A 00000002 00000003 0096  80   FFFFFFFF   │
                                               00000000   │
 _   0032  000012 00000003 00000004 0096  80   FFFFFFFF   │
                                               00001000   │
 _   003D  00001A 00000004 00000005 0096  80   FFFFFFFF   │
                                               00000000   │
 _   0048  000022 00000005 00000000 0096  80   FFFF3000   │
                                               00000000   │
 _   0053  00002A 00000006 00000007 0097  C0   FFFFFFFF   2
                                               00000000   │
 _   005E  000032 00000007 00000008 0097  80   FFFFFFFF   │
                                               00000000   │
 _   0069  00003A 00000008 00000009 0097  80   FFFFFFFF   │
                                               00000000   │
 _   0074  000042 00000009 0000000A 0097  80   FFFFFFFF   │
                                               00000000   │
 _   007F  00004A 0000000A 0000000B 0097  80   FFFFFFFF   │
                                               00000000   ┘
```

Elements of a mapped space map page for a segmented table space

1 Header showing segment size, number of segments covered, count of free segments, and IDs of the first free segment and last formatted page.

2 One line for each segment controlled by the space map page. Each line contains the following information:

- The offset of the segment entry within the page
- The number of the first page described by this segment entry
- The number of this segment
- The number of the next segment for this OBID
- The OBID itself
- Flags describing the status of the segment

Selects for a mapped space map page for a segmented table space

Select	Description
P	Display the first page in the segment whose entry you selected.
S	Return to the Page Display panel for the space map page and place the cursor at the space bits for the first page in the segment.
U	Return to the Page Display panel for the space map page and place the cursor at the update bits for the first page in the segment.

Commands for a mapped space map page for a nonsegmented table space

Command	Description
LOCATE	Positions the panel to the entry for the first segment at or following the specified page offset. The format is LOCATE *xxxx* where *xxxx* is the hexadecimal offset. LOCATE can be abbreviated as L.
NEXT	Change the subject of the Space Map Page panel to the next space map page.
OBID	Restricts the display to segments allocated to the OBID specified. The format is OBID *xxxx* where *xxxx* is the 4-character OBID. Specify **OBID 0000** to see only unallocated segments.
PREV	Change the subject of the Space Map Page panel to the previous space map page.
RESET	Expands the display after OBID has been specified, such that segments for all OBIDs and for unallocated segments are listed. RESET can be abbreviated as R.

Using function keys for the NEXT and PREV commands

If you have assigned function keys to the RIGHT and LEFT commands, you can use these keys to execute the NEXT and PREV commands respectively.

Using the MAP Command Against Data Pages

Overview

This unit provides information on using the MAP command against space map pages on a Page Display panel.

Restriction on the use of the MAP command

The MAP command applies only to table spaces.

Purpose of the command

The output from the MAP command can help resolve data integrity errors that !DB/SMU finds and identifies in an error message. The MAP command formats the data in a space map page for a segmented or nonsegmented table space. There is a different type of output for the:

- Header page
- Space map pages
- Data pages

Information about mapping header pages and space map pages is found in the preceding units.

Format of the command

The MAP command cannot be abbreviated.

Prerequisite to issuing the command

To issue the MAP command for a data page, you must first display a data page on the Page Display panel. DB2 data pages begin with page 2 for noncompressed spaces, or after the last dictionary page for compressed spaces.

Issuing the command

Once you have displayed a header page using the Page Display panel, follow this step to issue the command.

Step	Action
1	Perform one of the following: • Type **MAP** on the command line of the Page Display panel; or • Type **DO** on the command line of the Page Display panel. Then type the number of the command you want in the Select field of the pop-up menu.

Result of issuing the MAP command

For data pages, the MAP command decodes the page into its constituent header, rows, holes, and so forth.

Output of the MAP command against an uncompressed data page

The output of the MAP command against an uncompressed data page is shown in this example for a table space Data Page panel.

```
TABLE SPACE DATA PAGE -------   DB/SMU  ---  DB2=DSN2  ----------- LINE 1 OF 7
Cmd ===>                                                  Scroll ===> CSR
 Page = 2         (2)         Selects: S to set position, F to format an item.
PF23/PF22 = +/- 1      pages. Cmds: EXPAND All, or RESET Only Summary Data.
                                    NEXT Next or PREV Prior page; PRNT; DO.
    OFF
SEL SET         ITEM        HEX                                CHAR
--- ---- ----------------- --------------------------------  ----------------
 F  0000 ** PAGE HEADER **  00000000000000FF000002000F1D00DB *................*
 F  0014 ** ROW **          000048001501C2C2C2C2F0F0F0F28000 *......BBBB0002..*
 _  005C ** ROW **          000043001502C1C1C1C1F0F0F0F18000 *......AAAA0001..*
 _  009F ** ROW **          00003C001503C1C2C3C4C5C6C7C87FFF *......ABCDEFGH".*
 _  00DB ** FREESPACE **    00000000000000000000000000000000 *................*
 F  0FF8 ** ID MAP **       009F005C0014                     *...*..*
    0FFE ** PAGE END **     00C5                             *.E*
 ***************************** BOTTOM OF PAGE DATA ****************************
```

Output of the MAP command against a compressed data page

Partial output of the MAP command against a compressed data page is shown in this example for a Table Space Data Page panel. *Compressed rows cannot be formatted.*

```
TABLE SPACE DATA PAGE -------   DB/SMU  ---  DB2=D31A  -------- LINE 1  OF 13
Cmd ===>                                                    Scroll ===> CSR
 Page = 32        (20)        Selects: S to set position, F to format an item..
PF23/PF22 = +/- 1      pages. Cmds: EXPAND All, or RESET Only Summary Data.
                                    NEXT Next or PREV Prior page; PRNT; DO.
    OFF
SEL SET         ITEM        HEX                                 CHAR
--- ---- ------------------ --------------------------------  ----------------
 _  0000 ** PAGE HEADER **  10000000000000FF00002000083906C7 *..............G*
 _  0014 ** COMPRESSED ROW  04000C0037011B3BDCC405A1          *.........D..*
 _  0020 ** COMPRESSED ROW  04000C00370230FC3ACF95A1          *..........n.*
 _  002C ** COMPRESSED ROW  04000E003703599A3EB70C995A20      *...........r..*
 _
```

Selects for a table space data page

Select	Description
F	Format the selected item as follows: • A *row* is formatted into its constituent columns. An example panel illustrating row formatting is found in "Using the ROWS Command" on page 372. *The F select is not supported for compressed rows.* • An *ID map* is decoded into individual ID map entries. An example ID map is found later in this unit. • A *page header* is decoded into its constituent fields as found in "Using the MAP Command Against Header Pages" on page 355.
S	Return to the Page Display panel for the space map page and place the cursor at the space bits.

Output of an F select—an example ID map

This is an example ID map.

```
ID MAP ----------------------   DB/SMU  ---  DB2=DSN2  ---------- FIELD 1 OF 3
Cmd ===>                                                  Scroll ===> CSR
 PAGE  ITEM
   OFFSET    DATA              DESCRIPTION
 ----------  ----------------  ---------------------------------------------
 0FF8  0000  009F              ID Map Entry   3: Offset to NU34YIN.FQEFMXF
 0FFA  0002  005C              ID Map Entry   2: Offset to NU34YIN.FQEFMXF
 0FFC  0004  0014              ID Map Entry   1: Offset to NU34YIN.FQEFMXF
 ****************************** BOTTOM OF DATA ********************************
```

Commands for a table space data page

Command	Description
DO	Display the DO command menu.
EXPAND	Add column type and attribute values to the display.
NEXT	Change the subject of the Space Map Page panel to the next space map page.
PREV	Change the subject of the Space Map Page panel to the previous space map page.
PRNT	Send the entire contents of the page, not just the portion currently displayed, to a JES printer.
RESET	Reset the panel to display only summary information.

Using function keys for the NEXT and PREV commands

If you have assigned function keys to the RIGHT and LEFT commands, you can use these keys to execute the NEXT and PREV commands respectively.

Using the PRNT Command

Overview

This unit provides information on using the PRNT command when you are using a Page Display panel.

Purpose of the command

Use the PRNT command to send a copy of the contents of a DB2 page from a table space or index space *immediately* to a JES printer. The entire contents of the page are printed, not just those lines that can be displayed in a single screen display.

Format of the command

The PRNT command cannot be abbreviated. Options for the PRNT command are specified using the !DB/SMU Housekeeping User Options. To review the PRNT options, follow these steps:

Step	Action
1	Select **H.0** on the !DB/SMU Primary Menu.
2	Press Enter. **Result:** !DB/SMU displays the Housekeeping Profile Administration Menu.
3	On the Profile Administration Menu, select option 2, Configuration Information.
4	Press Enter. **Result:** !DB/SMU displays the Configuration Information Menu containing your PRNT options.

Issuing the command

Follow this step to issue the command.

Step	Action
1	Perform one of the following: • Type **PRNT** on the command line of the Page Display panel; or • Type **DO** on the command line of the Page Display panel. Then type the number of the command you want in the Select field of the pop-up menu.

Using the RESET Command

Overview

This unit provides information on using the RESET command when you are using a Page Display panel.

Purpose of the command

Use the RESET command to discard all modifications you have made to the currently displayed page during the current !DB/SMU repair session. The data displayed after the execution of RESET is the original version of the DB2 page.

Format of the command

The RESET command can be abbreviated as RES.

Issuing the command

Follow this step to issue the command.

Step	Action
1	Perform one of the following: • Type **RESET** on the command line of the Page Display panel; or • Type **DO** on the command line of the Page Display panel. Then type the number of the command you want in the Select field of the pop-up menu.

Contrasting RESET and CANCEL

In contrast to RESET, the CANCEL command discards only those changes made to the DB2 page since the last interrupt (for example, an Enter or PF key). For information about CANCEL, see “Using the CANCEL Command” on page 350.

Using the ROWS Command

Overview

This unit provides information on using the ROWS command when you are using a Page Display panel.

Restriction on use of the ROWS command

The ROWS command is not available for compressed rows.

Purpose of the command

Use the ROWS command or its function key from a Page Display panel to display the table rows contained in the table space page. The resulting Rows Display panel lists the rows of a page in offset order, the length of each row, and the ID map entry that points to the row.

```
ROWS DISPLAY -----------------   DB/SMU  ---  DB2=DSN2  ---------- ROW 1 OF 33
Cmd ===>                                                    Scroll ===> CSR
 Page = 9         (9)          Select:  S to set position in dump screen,
PF23/PF22 = +/- 1      pages.           D to delete a row, F to format a row.
                ID     ROW
SEL  OFFSET  MAP     LEN  TABLE NAME           DATA
---  ------ ---- -------  ------------------   --------------------------------
 _    0014     9      89  SYSTABLEPART         ..........QF81EPNNQF81TRAN..
 _    006D    10     100  SYSTABLESPACE        ........QF81EPNNCL12TXI QF81TRAN
 F    00D1     8     162  SYSTABLES            ......................B.......DO
 _    0173    37     134  SYSINDEXES           ...h...h..........XEPNNCLSQF81
 _    01F9    18      99  SYSCOLUMNS           .......h......DOM_PROPER_NAME..D
 _    025C    22      95  SYSCOLUMNS           ..............DOM_PLACARD..DOMES
```

Format of the command

The ROWS command cannot be abbreviated. You can use PF18 on a Page Display panel to perform the equivalent function to the ROWS command.

Issuing the command

Step	Action
1	Perform one of the following: • Type **ROWS** on the command line of the Page Display panel; • Type **DO** on the command line of the Page Display panel. Then type the number of the command you want in the Select field of the pop-up menu; or • Press the appropriate function key from the Page Display panel.

Selects available on the Rows Display panel

These selects are available on the Rows Display panel. You can select any number of rows in the page.

Select	Result of Using this Select
D	!DB/SMU marks the rows you select using the D select for deletion in the DB2 REPAIR job stream that it creates at the end of repair processing. You can mark any number of rows for deletion. If you leave this page and return, !DB/SMU maintains the deletions just as it does prototype changes to a page. If the table containing the deleted row is indexed, DB2 deletes the entries that point to this row from all the indexes that are dependent on the table.
F	!DB/SMU formats the data in the columns of the row you select using the F select and displays the result in the Formatted Row panel, described below.
S	!DB/SMU returns to the Page Display panel with the cursor positioned at the offset indicated on the Rows Display panel.

Example Formatted Row panel

This is an example Formatted Row panel. You can add column type and attributes (length-, null-, and default-specification) values to the panel by typing **EXPAND** on the command line. You can remove these values by typing **RESET** on the command line.

```
FORMATTED ROW ----------------   DB/SMU  ---  DB2=DSN2  ------- COLUMN 1 OF 39
Cmd ===>                                                      Scroll ===> CSR
 Page = 9         (9)      Cmds: EXPAND Add or RESET Remove column attributes.
                                 EDIT ON Allow or OFF Inhibit column editing.
PAGE ROW                                          SYSIBM.SYSTABLES
 OFFSET  COLUMN NAME         VALUE             HEX
-------- -----------------  ----------------  --------------------------------
00D1 000 *** ROW PREFIX *** *..s...*          0000A2001308
00D7 006 ** LINK POINTER ** *....*            0000098A
00DB 00A ** LINK POINTER ** *....*            0000098A
00DF 00E ** LINK POINTER ** *....*            00000912
00E3 012 ** LINK POINTER ** *....*            00000000
00E7 016 ** LINK POINTER ** *....*            00000925
00EB 01A ** LINK POINTER ** *..B.*            0012C207
00EF 01E ** LINK POINTER ** *....*            00000000
00F3 022 NAME               *..DOMESTIC*      0008C4D6D4C5E2E3C9C3
00FD 02C CREATOR            *QF81    *        D8C6F8F140404040
0105 034 TYPE               *T*               E3
0106 035 DBNAME             *QF81TRAN*        D8C6F8F1E3D9C1D5
010E 03D TSNAME             *QF81EPNN*        D8C6F8F1C4D6D4D4
0116 045 DBID                845              834D
0118 047 OBID                21               8015
011A 049 COLCOUNT            5                8005
011C 04B EDPROC             *        *        4040404040404040
```

Commands available on the panels associated with the ROWS command

These commands are available on the panels associated with the ROWS command.

Command	Available on Panel	Description
EXPAND	Header Page panel	Entering EXPAND on the command line adds column type and attribute values to the panel.
PRNT	All panels described in this unit	Entering PRNT on the command line causes the entire contents of the page, not just that portion currently displayed, to be immediately sent to a JES printer. See "Using the PRNT Command" on page 369.
RESET	Header Page panel	Entering RESET on the command line returns the panel to displaying only summary information.

Using the XKEYS Command

Overview

This unit provides information on using the XKEYS command when you are using a Page Display panel.

Restriction on the use of XKEYS

XKEYS and its synonyms apply only to DB2 pages for index spaces.

Purpose of the command

The XKEYS command on a Page Display panel formats and displays the current DB2 index space page. The panel displaying the formatted page can have one of the following titles depending on its contents:

- Root Page Keys
- Leaf Page Keys
- Non-Leaf Page Keys

In the balance of this unit, these panels are referred to as *pagetype* Page Keys panels.

Format of the command

Any of the following can be used as a synonym for XKEYS:

- The KEY command, and abbreviations K or >KEY

  Note: Using the command KEYS can cause a display of ISPF keys depending on your ISPF profile definition. Therefore, Candle suggests that you not use the command KEYS as a synonym for XKEYS.

- The ROWS command, which performs the same function as XKEYS for an index page

Issuing the command

Follow this step to issue the commands

Step	Action
1	Perform one of the following: • Type **XKEYS**, **KEY**, **ROWS**, or **K** on the command line of the Page Display panel; or • Type **DO** on the command line of the Page Display panel. Then type the number of the command you want in the Select field of the pop-up menu.

Other ways to display a Page Keys panel

In addition to typing a command on the command line, you can display a *pagetype* Page Keys panel in one of these ways:

- Pressing PF18 on the Page Display panel
- Accessing the online help menu for the Page Display panel and selecting the KEYS command

Elements of the Page Keys panels

This is an example *pagetype* Page Keys panel, a scrollable, formatted panel.

```
KEYS DISPLAY ----------------   DB/SMU  ---  DB2=DSN2  ---------- KEY 1 OF 140
Cmd ===>                                                    Scroll ===> CSR
 Page = 145       (91)  1      Select with S to set position in dump screen,
PF23/PF22 = +/- 1      pages.              P to show keys for next level.
     OFF  FIRST       ROWS       5 KEYLEN=26
SEL  SET  ROWID    INDEXED  KEY               KEY IN HEX
      2    3         4      6                    7
 --  ---- -------- -------  ----------------  --------------------------------
 _   002D 00008F         1  DXULALL QW48EXP3  C4E7E4D3C1D3D340D8E6F4F8C5E7D7F3
 _   004A 000092         1  DXULPRS QW48EXP3  C4E7E4D3D7D9E240D8E6F4F8C5E7D7F3
 _   0067 000093         1  DXULWEL QW48EXP4  C4E7E4D3E6C5D340D8E6F4F8C5E7D7F4
 _   0084 000094         1  D2COMIT QX32GET1  C4F2C3D6D4C9E340D8E7F3F2C7C5E3F1
 _   00A1 000095         1  EMPRRT  DT20HRIS  C5D4D7D9D9E34040C4E3F2F0C8D9C9E2
 _   00BE 000096         1  F30A00 XF30PLAN   C6F3F0C1F0F05BE7C6F3F0D7D3C1D540
 _   00DB 000097         1  F30A01 XF30PLAN   C6F3F0C1F0F15BE7C6F3F0D7D3C1D540
 _   00F8 000098         1  GATEUPDTENTSGATU  C7C1E3C5E4D7C4E3C5D5E3E2C7C1E3E4
 _   0115 000099         1  HABC    APSAPPB   C8C1C2C340404040C1D7E2C1D7D7C240
```

Elements of the Page Keys panels (continued)

1 Page number, in decimal and hexadecimal form, of this page within the index space.

2 Offset to the key record

3 RID of the row.

4 Number of records indexed

5 Key length

6 Value of the key shown in character mode

7 Value of the key in hexadecimal

Selects available on the Page Keys panels

The selects in the chart are supported for an index page. Both are requests to reformat the panel. If you mix the types of selects, !DB/SMU processes only the first select it encounters.

Select	Result of Using this Select
P	!DB/SMU displays the Keys Display panel for the page at which the selected entry points. Thus, if the current index page is a root page, the Keys Display is for a leaf page or non-leaf page. If the current index page is a leaf page, the Keys Display is for a table space page instead of an index page on the next level.
S	!DB/SMU returns to the Page Display panel with the cursor positioned at the offset indicated on the *pagetype* Page Keys panel.

Command available on the Page Keys panels

You can enter the PRNT command on the *pagetype* Page Keys panels. Entering PRNT causes the entire contents of the page, not just that portion currently displayed, to be immediately sent to a JES printer. See “Using the PRNT Command” on page 369.

Using !DB/SMU to Initiate DB2 Utilities

Chapter 20. Introduction to Using !DB/SMU to Initiate DB2 Utilities

Introduction

This chapter provides the background information you need to use !DB/SMU to run a subset of the DB2 utilities or to run OEM utilities for DB2.

Chapter Contents

What DB2 Utilities Can I Run?

Overview

This unit lists the DB2 utilities you can initiate using !DB/SMU and discusses profiles forrunning the DB2 utilities.

Background about using !DB/SMU to run utilities

You can use !DB/SMU to run specific DB2 utilities during either online or batch operations. You can also use !DB/SMU to run OEM utilties for DB2.

Profile for running a DB2 utility

!DB/SMU uses default settings to run the DB2 utilities unless you provide a utility profile. By defining a custom utility profile, you provide specifics for your site. Such a utility profile contains information about general options, restart options, and options for each of the utilities you want to run. Typically, you need to perform this operation only once. You can modify the profile if changes are required. You can also migrate a utility profile from release to release of !DB/SMU.

You can also define more than one utility profile. The profile that you are using for a given session is referred to as the *active* utility profile. For more information about defining and using utility profiles, see "General Options for Creating DB2 Utility Profiles" on page 393.

Utilities you can run using !DB/SMU

You can use !DB/SMU to run the following DB2 utilities or their equivalent OEM versions:

- COPY (for full or incremental image copies)
- MERGECOPY
- MODIFY
- QUIESCE
- RECOVER
- REORG
- REPAIR
- RUNSTATS

Available DB2 commands

You can use !DB/SMU to execute the following DB2 command:

- REBIND (for dependent plans and packages)

Background about Using !DB/SMU to Run DB2 Utilities

Overview

This unit summarizes some of the advantages you receive when you use !DB/SMU to run DB2 utilities.

Advantages of using a utility profile

A utility profile makes it easy to specify a variety of complex operating instructions for DB2 utilities. These options remain in effect until the active profile changes. You can quickly change the options for a job by making a different utility profile the active one.

The executable job streams

!DB/SMU builds a complete and executable job stream. The job stream includes the job statements, utility EXEC and DD statements, and the necessary utility control statements.

Submitting a utility job stream

An executable job stream for running DB2 utilities is ready for submission whenever you do any of the following:

- Execute the batch MONITOR command
- Make the appropriate selections from the online monitor exception report
- Execute the batch SCAN or SCANR command

Options for job execution

You can choose to execute a utility in one of these ways:

- A single step
- Separate steps for each object (table space or index space)
- Separate jobs for each object

This flexibility enables you to continue to process utility steps even if one step ABENDs.

Automatic generation of DD statements

!DB/SMU automatically generates DD statements that are tailored to your installation standards. !DB/SMU uses symbolic substitution variables to generate dataset names that fit your requirements. The dataset names are unique for each table space or index space being processed. Information about symbolic substitution variables can be found in "Substituting Values for Variables in the Skeleton Job Stream" on page 387.

Automatic calculation of DASD space

!DB/SMU *dynamically* adjusts the DASD space allocation parameters for the work files, copy output data sets, and unload datasets to the size of the dataset being processed. This prevents SB37 ABENDs caused by too small an allocation and also reduces wasteful overallocations done to avoid such abends.

Automatic recall of migrated datasets

!DB/SMU detects migrated and missing datasets before !DB/SMU generates any utilities. You can choose to recall automatically any migrated datasets.

General Information about Using Utility Profiles

Overview

This unit provides general information for using a DB2 utility profile.

Use of the active utility profile

!DB/SMU uses the specifications in the active utility profile to generate JCL for the DB2 utilities. The specifications apply to both online and batch jobs.

Viewing a list of existing utility profiles

To view the list of existing utility profiles, type **5.4** to access the Utility Profiles panel through the Profiles option of the !DB/SMU Primary Menu.

Setting an active utility profile

You can set the active profile (make a specific profile the active one) in any of the following ways:

- For batch jobs, you can specify the name of the utility profile on the Generate Batch Job panel. !DB/SMU displays this panel whenever you build a monitor run profile or scan run profile.
- When generating a job stream by selecting an online monitor exception report, you can enter the **UTIL OPTS <***profilename***>** command on the Monitor Exception Report panel. If you omit the *profilename* parameter, !DB/SMU displays a menu that provides selection list of profiles.

Substituting Values for Variables in the Skeleton Job Stream

Overview

This unit discusses the substitution symbols that !DB/SMU provides for such variables as date, time, user ID, database name, and so forth.

Background about substitution variables

!DB/SMU automatically generates a job stream based on definitions in the active utility profile and any other profiles that are associated with the job. To do this, !DB/SMU uses skeletons to build jobs for each of the utilities. The skeletons contain an extensive list of substitution symbols for the variables in the job stream. You can view a complete, commented list of these symbols in the utility skeleton members in the SKELS dataset.

Examples of substitution symbols for common variables in a job

Review the chart for a description of some of the symbols that !DB/SMU substitutes for common variables.

Symbol	Substituted Value
&DBNAME	Database name
&DATE	Current date (*yyddd*)
&DATE2	Current date (*yymmdd*)
&DBID	DB2 ID
&SPACENM	Name of the table space or index space
&SYSUID/&USERID	Current TSO User ID
&TIME	Current time (*hhmmsst*)
&UTILID/&UID	Utility ID
&VCATNAME	VSAM catalog name

Initiating a DB2 Utility during Online Operation

Overview

This unit points to the general procedures for initiating a DB2 utility during an online session.

Restriction on creating JCL to execute a DB2 utility during online operation

During online operation, you can create the JCL necessary to execute DB2 utilities. This can be done only from the monitor exception report panel of the !DB/SMU monitor function. For instructions, refer to “Options for Running the DB2 Utilities from the Monitor Exception Report” on page 310.

Initiating a DB2 Utility from a Batch Job

Overview

You can use a monitor run profile or scan run profile to create a batch job to initiate DB2 utilities. This unit explains the procedures to do so.

Methods for initiating DB2 utilities in a batch job

You can initiate DB2 utilities in a batch job by using any of the following methods:

- Use a monitor run profile to initiate a DB2 utility.
- Use a scan run profile to name an exception profile. The exception profile initiates the DB2 utility.
- Use either a monitor run profile or a scan run profile to name an action profile. The action profile initiates one or more DB2 utilities.

Creating batch procedures to initiate DB2 utilities

To understand how to use these methods during batch operations, see "Generating Batch Jobs" on page 469.

Using OEM Utilities for DB2

Overview

You can use !DB/SMU to run OEM utilities for DB2. This unit explains these procedures.

Preparing to use an OEM utility

Complete these initial preparations to use an OEM utility. These preparations use standard ISPF skeleton processing. Candle strongly suggests that you make and save backup copies of any skeletons you plan to alter.

Step	Action
1	Make a copy of the *hilev*.SKELS dataset.
2	Use ISPF edit to open the KTUUTJCL skeleton.
3	Locate the job statements that provide support for the OEM utility you are using.
4	Edit the job and save the edited member in the SKELS library using a name of your choice.
5	Edit any additional skeletons as needed to modify product-specific parameters that must be customized for the OEM utility.

Creating a utility profile for the OEM utility

You must define a utility profile to provide operational instructions for the OEM utility. Use the procedures explained in “General Options for Creating DB2 Utility Profiles” on page 393 and perform these additional tasks.

Task	Procedure
Identify the skeleton member you created for the OEM utility.	On the General Options panel, type the name of the skeleton member in the Model JCL field. See “Generating the batch job stream” on page 470.
On the User Variables panel, define any variables that are needed for the OEM utility.	See “Defining User Variables” on page 412 for details.

Initiating an OEM utility

Use the following procedures when you initiate the OEM utility from the online monitor facility or a batch job.

Step	Action
1	Follow procedures to display the Generate Batch Job panel.
2	Use the Tab key to move to the Utility Profile field. Type the name of the utility profile that you created for the OEM utility.
3	Use the Tab key to move to the remaining entry fields and complete them as needed.
4	Press Enter. **Result**: !DB/SMU displays the Process Member panel.
5	Choose an appropriate option for the job.

Managing maintenance for !DB/SMU

Before using OEM utilities for DB2, you should make copies of the *hilev*.SKELS and *hilev*.PANELS libraries and modify the applicable members within these libraries. This procedure preserves your customization through the maintenance process. After you apply maintenance and refresh the product libraries, determine if you need to refresh your copies of these libraries. Refer to the documentation that shipped with the maintenance tapes to evaluate this need.

Managing maintenance for OEM utilities

If your OEM utilities change, you should check any procedures you have modified to determine if they need to be updated. The following chart shows the type of changes that could effect the procedures you have modified.

Type of Change	!DB/SMU Procedure
Program name	In *hilev*.SKELS(KTUUTJCL)
Load library requirements	In *hilev*.SKELS(KTUUTJCL)
Product-specific utility datasets, for example, names, additions, and eliminations	In *hilev*.SKELS(KTUUTJCL, KTUUTDD*) Check the JCL generation procedure for each utility for the apprpriate member names.

Managing maintenance for OEM utilities (continued)

Type of Change	!DB/SMU Procedure
Control statements	In *hilev*.SKELS(KTUUT*xx*) Check the JCL generation procedure for each utility for the apprpriate members to modify. In *hilev*.PANELS(SFXPUTUS) for names of user variables

Chapter 21. General Options for Creating DB2 Utility Profiles

Introduction

This chapter and the two that follow contain information about creating a profile for the DB2 utilities. This chapter discusses the use of the general options panels that apply to both table spaces and index spaces. The following chapters discuss panels specific to table spaces and index spaces respectively.

Chapter Contents

Background about DB2 Utility Profiles

Overview

This unit provides background about using !DB/SMU to interactively create a utility profile for the DB2 utilities and points you to source information.

Background about creating a utility profile

!DB/SMU gives you the ability to execute specific DB2 utilities during batch and online operations. The execution of a utility requires information you specify by means of a utility profile. You create a utility profile tailored to your needs by using a series of panels that you access from !DB/SMU Utility Profiles menu.

Considerations when initiating DB2 utilities in batch

After you have defined your DB2 utility profiles, you can execute the utilities in either online or batch mode. This chart shows the additional profiles required when you initiate DB2 utilities in batch mode and refers you to appropriate chapters in the Batch Processing section of this guide for additional information.

DB2 Utilities That Can Be Initiated From Batch Jobs	**Action Profile, p. 557**	**EXCPTCTL Stmts, p.529,p.576**	**Exception Profile, p. 509**	**Monitor Run Profile, p. 491**	**Scan Run Profile, p. 491**
COPY (full/ incremental)	√	√	√	√	
MERGECOPY	√	√			
MODIFY	√	√			
QUIESCE	√	√			
REBIND for dependent plans and packages	√	√			
RECOVER	√	√	√	√	
REORG	√	√	√	√	
RUNSTATS	√	√	√	√	√

Organization of information to create a utility profile

!DB/SMU provides a series of 22 panels with which you can interact to specify values for utility profile options. The information you need to use these panels is structured as shown in this chart:

Task You Want to Perform	Source of Information
Provide values for general options that apply to both table spaces and index spaces.	This chapter
Provide values for options that apply to table spaces	"Options for Creating DB2 Utility Profiles for Table Spaces" on page 415
Provide values for options that apply to index spaces	"Options for Creating DB2 Utility Profiles for Index Spaces and BIND/REBIND Plans and Packages" on page 441
Review additional detailed information about all of the panels and their fields	Online help for each panel
Review a detailed description of the DB2 utilities and their options	*IBM DATABASE 2 Command and Utility Reference*

Beginning DB2 Utility Profile Definitions

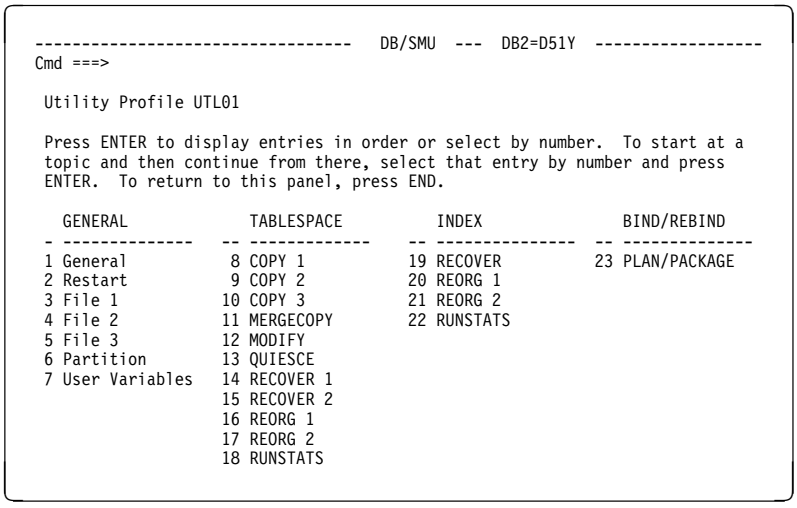
Overview

This unit contains procedures for accessing and using the profile options panels.

Creating a new DB2 utility profile

Follow these procedures to create a new DB2 utility profile.

Step	Action
1	On the !DB/SMU Primary Menu, enter the following fast path command in the Option field: • **5.4**
2	Press Enter. **Result**: !DB/SMU displays the Utility Profiles panel.
3	Type a name for the new profile in the Enter Name Here to Create a New Utility Profile field. This name is carried forward in the field shown in this chapter as *profilename*.
4	Press Enter. **Result**: !DB/SMU displays the Utility Profile *profilename* menu. (In the balance of the three chapters that support the creation of DB2 utility profiles, this panel is referred to as the *Utility Profile menu.*)



```
---------------------------------  DB/SMU  ---  DB2=D51Y  ------------------
Cmd ===>

 Utility Profile UTL01

 Press ENTER to display entries in order or select by number.  To start at a
 topic and then continue from there, select that entry by number and press
 ENTER.  To return to this panel, press END.

   GENERAL             TABLESPACE          INDEX               BIND/REBIND
 - --------------   -- -------------    -- ---------------  -- --------------
 1 General           8 COPY 1           19 RECOVER          23 PLAN/PACKAGE
 2 Restart           9 COPY 2           20 REORG 1
 3 File 1           10 COPY 3           21 REORG 2
 4 File 2           11 MERGECOPY        22 RUNSTATS
 5 File 3           12 MODIFY
 6 Partition        13 QUIESCE
 7 User Variables   14 RECOVER 1
                    15 RECOVER 2
                    16 REORG 1
                    17 REORG 2
                    18 RUNSTATS
```

Creating a new DB2 utility profile (continued)

Step	Action
5	To begin your profile definition with the first panel on the list, type **1** on the command line.
6	Press Enter. **Result**: !DB/SMU displays the General Options panel.
7	Complete the fields on the General Options panel.
8	Press Enter to display the next definition panel. **Result**: !DB/SMU displays the Restart Options panel.
9	Repeat steps 7 and 8 to complete the remaining definitions.

Changing a DB2 utility profile

Follow these procedures to change an existing utility profile.

Step	Action
1	On the !DB/SMU Primary Menu, type **5.4** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the Utility Profiles panel that lists existing profiles.
3	Type **S** in the Sel field of the named profile you want to change.
4	Press Enter. **Result**: !DB/SMU displays the Utility Profile *profilename* menu.
5	Type the number of the definition you want to change.
6	Press Enter. **Result:** !DB/SMU presents the appropriate options panel. Update the definition as needed and save your changes.
7	Type **END** on the command line or press the appropriate function key to exit the panel and return to the Utility Profile *profilename* menu.
8	Repeat steps 5 through 7 to change additional definitions.

Including DB2 Commands in Your Utility Job Stream

Overview

This unit discusses the use of the options CMDS BEFORE and CMDS AFTER. These options permit you to generate START and STOP DATABASE commands as part of your utility profile.

Background about the command options

Many utility option panels include the options that permit you to include DB2 -START DATABASE or -STOP DATABASE commands as the first and last steps of the job that executes the given DB2 utility.

Parameters for the command options

When you include the options CMDS BEFORE and CMDS AFTER in your utility profile definition, you can specify the parameters as shown in these lists.

CMDS BEFORE

Generates a DB2 command in the job step that immediately precedes the utility steps. You can specify one of the following parameters for this command option:

P Generates the -STOP DATABASE command for each space to be processed. Caution—most DB2 utilities require that the table space or index space *not* be stopped.

R Generates the -START DATABASE ACCESS(RO) command for each space to be processed. RO permits read only access to the database.

U Generates the -START DATABASE ACCESS(UT) command for each space to be processed. UT allows only DB2 utilities to process the spaces.

V Equivalent to P + U. Generates -STOP DATABASE first, then -START DATABASE ACCESS(UT).

Parameters for the command options (continued)

CMDS AFTER

Generates a DB2 command as a last job step that immediately follows the utility steps. You can specify one of the following parameters for this command option:

P Generates the -STOP DATABASE command for each space to be processed.

R Generates the -START DATABASE ACCESS(RO) command for each space to be processed. RO permits read only access to the database.

W Generates the -START DATABASE ACCESS(RW) command for each space to be processed. RO permits both read and write access to the database.

Z Equivalent to P + R. Generates -STOP DATABASE first, then -START DATABASE ACCESS(RO).

General Options for a Utility Profile

Overview

This unit contains information on defining the general job options.

Purpose

The general options establish the basic JCL information for submitting a job.

Access

Select option 1 on the Utility Profile panel to directly access the General Options panel.

Elements of the General Options panel

This is the General Options panel.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  ----------------------
Cmd ===>                                                      mm/dd/yy 10:48


                    G E N E R A L   O P T I O N S

 1 DB2 SYSTEM ===>                DB2 Subsystem Identifier.

    2 STEPLIB ===> 'PP.DB2.V2R3.DSNLOAD'                 DB2 Load Library
                              Use quotes if not prefixed with your ID.
  3 MODEL JCL ===> KTUUTJCL    Utility Skeleton member name.
     4 REGION ===> 409K        Region size for EXEC including K or M.
  5 JOB CONTROL => 1           1 for all DB2 spaces in one Job Step.
                               2 if each DB2 space in a separate Job Step.
                               3 for each DB2 space in separate Job.

    6 MIGRATED ===> I          S to skip, I to include.
 7 SYSOUT CLASS => *           Class for JES output.

 8 JOB STATEMENTS:
===> //USERID   JOB (BHOU),UTILITY,CLASS=A,NOTIFY=TDDB31,
===> //         MSGCLASS=X
===> //*
===>
```

Elements of the General Options panel (continued)

1 DB2 subsystem identifier; contains the name of the DB2 subsystem you are currently using with !DB/SMU.

2 dataset name of the DB2 load library. Type the full name, enclosed in single quotes, unless the name begins with your user ID. This library is used as the STEPLIB in the generated JCL.

3 Name of the member in the !DB/SMU SKELS PDS that contains the model JCL and the substitution symbol values for your definitions. The Candle-supplied member is KTUUTJCL.

4 Size for the execution region in kilobytes or megabytes.

5 Processing options for the datasets. Determines whether processing of your generated job comprises a single job step, multiple job steps, or multiple jobs:

- **1** Processes all the datasets in a single step. The work file allocations (SYSUT1) accommodate the largest table space or index space. This choice offers the advantage of less overhead, but if the utility fails to process any object, then the entire DB2 utility job fails.
- **2** Processes each dataset in its own step. The utility can fail to process an object, but processing of the other objects continues if the model JCL specifies COND=EVEN on the EXEC statement.
- **3** Processes each space in a new job for partition independence. To make each job unique, append an &J to the job name specified in the JOB STATEMENTS section. (In this example, the job name would be entered as **BHOU&J**.)

6 Processing options for migrated datasets:

- **S** Skips all migrated datasets.
- **I** Includes all migrated datasets.

7 Class for JES output

8 Job information specific to your site

Restart Options for a DB2 Utility

Overview

This unit contains information on defining a restart option for a DB2 utiltity.

Purpose

The restart options permit you to specify whether or not a utility is to be restarted, and if it is to be restarted, whether to restart at a commit point or end-of-phase.

Access

Select option 2 on the Utility Profile menu to directly access the Utility Restart Options panel. Access this panel in sequence by pressing Enter on the General Options panel.

Elements of the Utility Restart Options panel

This is the Utility Restart Options panel.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A                  CSR
Cmd ===>                                                    mm/dd/yy 10:48

               U T I L I T Y   R E S T A R T   O P T I O N S

 1 UTILITY-ID ===>            Unique Utility ID, default is Userid.Jobname.

      2 START ===>            Leave blank if not restarting this Utility.
                              Else, restart at C Commit or P Phase.
```

Elements of the Utility Restart Options panel (continued)

1 Unique identifier necessary for each DB2 utility; sets the value of UID in the JCL EXEC statement.

2 Restart option for the utility; sets the value of UTPROC in the JCL EXEC statement.

C	Restarts at the last commit point
P	Restarts at the beginning of the last executed phase
blank	Does not restart the utility

General File Options for the REORG Utility

Overview

This unit contains information on defining the work, unload, and sort files.

Purpose

This panel permits you to specify the characteristics of the work files, unload files, and sort files used for reorganization.

Access

Select option 3 on the Utility Profile menu to directly access the REORG Utility Options panel. Access this panel in sequence by pressing Enter on the Utility Restart Options panel.

Elements of the Reorg Utility Options panel

This is the Reorg Utility Options panel.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  ----
Cmd ===>                                                 mm/dd/yy 10:54


                  R E O R G   U T I L I T Y   O P T I O N S

 WORK Files
     1 DDname ===> #          # = WORKnnnn, blank = "spacenm", default = SYSUT1.
       2 UNIT ===> SYSDA      Device type for Work files.
       3 SIZE ===> 101        Override Work dataset allocation value.
    4 STORCLAS===> S1           5 DATACLAS===> D1           6 MGMTCLAS===> M1

 UNLOAD Files
     1 DDname ===> #          # = UNLDnnnn, blank = "spacenm", default = SYSREC.
       2 UNIT ===> SYSDA      Device type for Unload datasets.
       3 SIZE ===> 102        Override Unload dataset allocation value.
    4 STORCLAS===> S2           5 DATACLAS===> D2           6 MGMTCLAS===> M2

 SORTOUT Files
     1 DDname ===> #          # = SOUTnnnn, blank = "spacenm", default = SORTOUT.
       2 UNIT ===> SYSDA      Device type for SORTOUT files.
       3 SIZE ===> 103        Override SORTOUT dataset allocation value.
    4 STORCLAS===> S3           5 DATACLAS===> D3           6 MGMTCLAS===> M3
```

Elements of the Reorg Utility Options panel (continued)

1 Names of the WORK, UNLOAD, and SORTOUT datasets:

- To use the name from the skeleton JCL for the work dataset, type **SYSUT1**. This option is recommended for the work DDname.
- To use the name from the skeleton JCL for the unload dataset, type **SYSREC**. This option can result in errors for the unload DDname if you run the utility for more than one table space and specify **ONE STEP=Y**.)
- To use the name from the skeleton JCL for the sort output dataset, type **SORTOUT**.
- To use the table space or index space name as the DDname, leave the field blank.
- To generate DDnames sequentially as WORK0001, UNLD0001, or SOUT0001, type **#**. (This option is recommended for UNLDDDNM and SOUTDDNM.)

If you use the utility skeleton JCL and type **#** for the DDname or leave the field blank, you must modify the JCL repeating section to use the correct DDnames.

2 Name of the device type of the dataset. The default is SYSDA.

3 Primary space allocation for the dataset:

- If you leave this field blank, !DB/SMU computes the required size and eliminates the problem of correctly estimating it.
- If you specify the size, then no size calculations are performed.

4 Storage attributes for the utility datasets

5 Data class for a new dataset

6 Management class for an SMS-managed dataset

General File Options for the COPY Utility

Overview

This unit contains information on defining local datasets for image copies.

Purpose

These options identify the primary and backup datasets for the DB2 COPY utility.

Access

Select option 4 on the Utility Profile menu to directly access the Copy Options panel for local copy files. Access this panel in sequence by pressing Enter on the Reorg Utility Options panel.

Elements of the Copy Options panel

This is the Copy Options panel.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  ----
Cmd ===>                                                    yy/mm/dd

                     C O P Y    O P T I O N S

 COPY Files (for the primary and backup copy at the LOCAL site)
    1 DDname1 ===>            # = COPYnnnn, blank = "spacenm", default =
      2 UNIT1 ===> SYSDA      Device type for Copy file 1.
      3 SIZE1 ===>            Override Copy dataset size (tracks).
   4 STORCLAS ===>              5 DATACLAS===>             6 MGMTCLAS===>
      7 STACK ===> N          All datasets on one tape? Y/N

    1 DDname2 ===>            # = COPYnnnn 1
      2 UNIT2 ===> SYSDA      Device type for Copy file 2.
  4. STORCLAS ===>              5 DATACLAS===>             6 MGMTCLAS===>
      7 STACK ===> N          All datasets on one tape? Y/N
```

Elements of the Copy Options panel (continued)

1 Name of the COPY dataset that becomes the DDname for the primary and backup datasets at the local site:

- To use the DDname from the utility skeleton JCL, type **SYSCOPY**.
- To use the table space or index name as the DDname, leave the field blank.
- To generate names sequentially in the form COPY0001, COPY0002, and so forth, type **#**.

2 Name of the device types for the primary and backup data set work files on both the local and recovery sites. The default is SYSDA.

3 Primary space allocation for the primary local dataset:

- If you leave this field blank, !DB/SMU computes the required size and eliminates the problem of correctly estimating it.
- If you specify the size, then no size calculations are performed.

4 Storage attributes for the datasets

5 Data class for a new dataset

6 Management class for an SMS-managed dataset

7 Storage option for copied datasets; you *must* enter a value in this field:

Y Store all the datasets on one tape
N Do not store all the datasets on one tape

General File Options for the RECOVERY Utility

Overview

This unit contains information on defining local datasets for the DB2 RECOVERY utility.

Purpose

These options identify the primary and backup datasets at the recovery site.

Access

Select option 5 on the Utility Profile menu to directly access the Copy Options panel for copy files at the recovery site. Access this panel in sequence by pressing Enter on the local Copy Options panel.

Elements of the Copy Options panel for recovery files.

This is the Copy Options panel for copy files at the recovery site.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  ----
Cmd ===>                                                    mm/dd/yy

                     C O P Y    O P T I O N S

 COPY Files (for the primary and backup copy at the RECOVERY site)
    1 DDname1 ===>             # = RECOnnnn
      2 UNIT1 ===> SYSDA       Device type for Recovey Dataset 1.
  3 STORCLAS ===>                4 DATACLAS===>              5 MGMTCLAS===>
      6 STACK ===> N           All datasets on one tape? Y/N

    1 DDname2 ===>             # = RECOnnnn 1
      2 UNIT2 ===> SYSDA       Device type for Recovery Dataset 2.
  3 STORCLAS ===>                4 DATACLAS===>              5 MGMTCLAS===>
      6 STACK ===> N           All datasets on one tape? Y/N
```

Elements of the Copy Options panel for recovery files.

1 DDnames of the primary and backup datasets respectively. To generate names sequentially in the form RECO0001, RECO0002, and so forth, type **#**.

2 Name of the device types for the primary and backup files respectively. The default is SYSDA.

3 Storage attributes for the utility datasets

4 Data class for a new dataset

5 Management class for an SMS-managed dataset

6 Storage option for copied datasets; you *must* enter a value in this field:

Y Store all the datasets on one tape
N Do not store all the datasets on one tape

Options for Partitioned Spaces

Overview

This unit contains information about the processing options for partitioned spaces.

Purpose

Use this panel to define what partitions in a partitioned space should be processed, and whether all partitions should be processed together or each partition should be processed separately. !DB/SMU ignores these specifications if it processes a non-partitioned space.

Access

Select option 6 on the Utility Profile menu to directly access the Partition Specifications panel. Access this panel in sequence by pressing Enter on the recovery site Copy Options panel.

Elements of the Partition Specifications panel

This is the Partition Specifications panel.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  ----------------------
Cmd ===>                                                      mm/dd/yy 10:48


              P A R T I T I O N   S P E C I F I C A T I O N S

    1
 DSNUM/PART ===> ALL    ALL to process all partitions together, or
                        EACH to process each partition separately, or
                        SOME to process the partitions identified below, or
                        a number to process the corresponding partition.
                  (This specification is ignored for non-partitioned spaces.)


If SOME, use any non-blank character to select the partitions to be processed.

 _  1     _  2     _  3     _  4     _  5     _  6     _  7     _  8     _  9     _ 10
 _ 11     _ 12     _ 13     _ 14     _ 15     _ 16     _ 17     _ 18     _ 19     _ 20
 _ 21     _ 22     _ 23     _ 24     _ 25     _ 26     _ 27     _ 28     _ 29     _ 30
 _ 31     _ 32     _ 33     _ 34     _ 35     _ 36     _ 37     _ 38     _ 39     _ 40
 _ 41     _ 42     _ 43     _ 44     _ 45     _ 46     _ 47     _ 48     _ 49     _ 50
 _ 51     _ 52     _ 53     _ 54     _ 55     _ 56     _ 57     _ 58     _ 59     _ 60
 _ 61     _ 62     _ 63     _ 64
```

Elements of the Partition Specifications panel

1 Specifies partition processing in the following way:

- To process a single partition, type its partition number.
- To process all of the partitions as a group, type **ALL**.
- To process each partition separately, type **EACH**.
- To process some of the partitions, type **SOME**, and then type any character next to the partitions you want to process.

Considerations for processing partitions

Note the following considerations for processing partitions:

Processing Choice	Consideration	Additional Information
One partitioned space	Produces only one set of output datasets	To assure unique data storage locations, include a discriminator for the output dataset names in the utility skeleton JCL.
Partitioned and non-partitioned spaces together	If you specify **EACH** or **SOME** as the DSNUM/PART, do not use a discriminator such as PART# for the output dataset names.	
Partitioned spaces individually	If you specify **EACH** for a utility that creates multiple output datasets (such as COPY, SORTOUT, or UNLOAD), an output dataset is created for each partitioned space.	To prevent overwriting data during processing, include a discriminator (such as PART#) in the utility's skeleton JCL (STEP JCL).
Multiple partitioned spaces together	If you specify **SOME** for a utility that creates multiple, output datasets (such as COPY, SORTOUT, and UNLOAD), two or more output datasets are created for each partitioned space.	To prevent overwriting data during processing, include a discriminator (such as PART#) in the utility's skeleton JCL (STEP JCL).

Defining User Variables

Overview

This unit contains information on defining customized user variables and overriding your existing definitions. Defining user variables is optional. You can bypass this option if you choose. If you choose to define customized user variables, these substitution variables become unique to your installation.

Purpose of user variables in a utility profile

User variables enable you to substitute specific values for substitution symbols in your JCL and control card statements. As !DB/SMU processes utility profile statements, it substitutes the user variable value you have specified whenever it encounters the corresponding substitution symbol. User variables are represented by the symbol &USERnn where *nn* can be a number from 01 through 62.

Example of the use of user variables

Assume your site uses a dataset naming convention that identifies a dataset with both a data type (TEST and PROD) and a copy number. User variables permit you to build this information into a DB2 utility profile. You can do this by assignent of user variables on the User Variables panel as follows:

```
TEST001 ===> &USER01
TEST002 ===> &USER02
PROD001 ===> &USER03
PROD002 ===> &USER04
```

Whenever !DB/SMU executes the associated utility profile, it substitutes the value in the left hand column whenever it encounters the substitution symbol in the right hand column.

Prerequisites for using user variables in a utility profile

You define user variables initially by modifying member SFXPUTUS in the Panels library. Follow these steps to modify member SFXPUTUS.

Step	Action
1	Using the ISPF editor, access member SFXPUTUS in the PANELS library. Review the comments and complete instructions contained in this member.
2	In the)INIT section, remove the RESP = ENTER line.
3	In the)INIT section, provide a definition of up to 32 characters for the &USER01 variable.
4	Define any additional variables (&USER02, &USER03, and so forth) that you require. The total number of user variables you can define is 63.
5	Remove the comment codes as indicated in the member.
6	In the)PROC section, provide a VPUT definition for each of the variables that you have defined.
7	Save your changes and end the editing session.

Access

Select option 7 on the Utility Profiles menu to directly access the User Variables panel. Alternatively, you can access this panel in sequence by pressing Enter on the Partition Options panel.

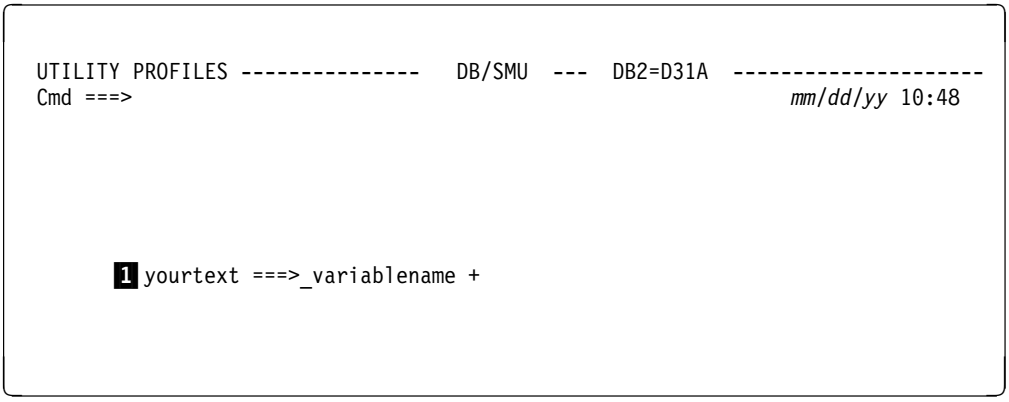
Overriding user variable definitions with the User Variables panel

When you access the User Variables panel, it displays all of the customized definitions you provided in SFXPUTUS. You can use the User Variables display to do the following:

- Add variables that you want for this utility profile
- Exclude unwanted variables from this utility profile
- Modify any of the definitions that you provided in SFXPUTUS

Elements of the User Variables panel

This is an example User Variables panel.



1 Format for entry of a user variable. Up to 62 user variables can be entered for a given utility profile.

Implementing user variables for a utility profile

After you complete the prerequisites and select the definitions for this utility profile, do the following:

Step	Action
1	Make a copy of the KTUUTJCL skeleton and any individual utility skeletons assign them names of your choice.
2	Add the &USER01 substitution variable and any others you want to the list of substitution variables in the first part of the skeleton copy.
3	Modify the control card statements in the second part of the skeleton copy to include these substitution symbols as it suits your needs.
4	Provide the name of this alternative skeleton as the name of the model JCL member in the General Options panel for this utility profile.

Chapter 22. Options for Creating DB2 Utility Profiles for Table Spaces

Introduction

This chapter contains information about creating profiles for the DB2 utilities. It discusses the use of the panels that apply to table spaces. The preceding chapter discussed the use of the general option panels that apply to both table spaces and index spaces. The following chapter discusses panels specific to index spaces.

Chapter Contents

Preparing to Create Utility Profiles

Overview

This unit provides information about using a series of panels to create a utility profile required to run a DB2 utility.

Background about creating a utility profile

!DB/SMU gives you the ability to execute specific DB2 utilities during batch and online operations. The execution of a utility requires information you specify by means of a utility profile. You can create a utility profile tailored to your needs by using a series of panels that you access from the Utility Profiles menu.

Organization of information to create a utility profile

!DB/SMU provides a series of 22 panels with which you can interact to specify values for utility profile options. The information you need to use these panels is structured as shown in this chart:

Task You Want to Perform	Source of Information
Provide values for general options that apply to both table spaces and index spaces.	"General Options for Creating DB2 Utility Profiles" on page 393
Provide values for options that apply to table spaces	This chapter
Provide values for options that apply to index spaces	"Options for Creating DB2 Utility Profiles for Index Spaces and BIND/REBIND Plans and Packages" on page 441
Review additional detailed information about all of the panels and their fields	Online help for each panel
Review a detailed description of the DB2 utilities and their options	*IBM DATABASE 2 Command and Utility Reference*

Options for Taking Image Copies of Table Spaces

Overview

This unit contains information on defining options for taking image copies of table spaces.

Purpose

Use this panel to specify the first and last steps for the DB2 COPY utility, share access for the table space, and the type of image copy you want to take.

Access

Select option 8 on the Utility Profile menu to directly access the Copy Utility Options panel. Access this panel in sequence by pressing Enter on the User Variables panel.

Primary substitution value for the COPY utility

The substitution variable &CPTRKS2 is the substitution variable used in the distributed skeleton JCL to contain the value that indicates the number of changed pages. Since the space allocation for the copy output in the model JCL is in 8K blocks, the number of blocks to which !DB/SMU writes copy output will be half the number of pages changed. If you change the output image copy blocksize, you must adjust the space allocation to accommodate the number of pages per output block.

Elements of the Copy Utility Options panel.

This is the Copy Utility Options panel.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D51Y  ----------------------
Cmd ===>                                                      yy/mm/dd 10:52

                      COPY UTILITY OPTIONS


1 CMDS BEFORE ===> V  P = Stop, R = Start RO, U = Start UT, V = P and U

2 CMDS AFTER  ===> Z  P = Stop, R = Start RO, W = Start RW, Z = P and R

3 SHRLEVEL    ===> R  C = CHANGE, R = REFERENCE

4 FULL        ===> H  Y = full copy
                      N = Incremental Image Copy
                      M = read Space Map Pages for Incremental Copy sizing
                      H = CHANGELIMIT
                      D = DFHSM CHANGELIMIT

   PERCENT 1  ===> 10 CHANGELIMIT percent 1 value (0 - 100)
   PERCENT 2  ===> 50 CHANGELIMIT percent 2 value (0 - 100)

  REPORTONLY   ===> N   Y = report only (a copy will not be made)
```

1 DB2 commands option; see “Including DB2 Commands in Your Utility Job Stream” on page 398.

2 DB2 commands option; see “Including DB2 Commands in Your Utility Job Stream” on page 398.

3 Establishes the type of share access to the table space while the Copy utility is running.

- **R** Allows only references to the table space while the Copy utility is running
- **C** Allows updates to and deletes from the table space while the Copy utility is running.

4 Specifies the type of image copy:

- **Y** Copies all the pages for a full image copy.
- **N** Copies only those pages that have been modified since the previous image copy. Produces an incremental copy.
- **M** Creates an incremental copy with accurate page sizes. !DB/SMU reads the DB2 space map pages of each dataset to determine the exact number of pages updated since the last image copy. *To use this option, you must have proper authorization to open and read the DB2 table space.*
- **H** Activates the CHANGELIMIT option (DB2 5.2 or higher only).

D Activates the DFSMS CHANGELIMIT option (DB2 5.2 or higher only).

PERCENT 1&2

These options are inactive unless CHANGELIMIT is active. The values range from 0-100 and represent the percentage of pages that have changed since the last copy was made. An incremental copy will be generated if the percentage of changed pages falls within the range you specify. A full copy will be generated if the percentage of changed pages exceeds the maximum value you specify.

Datasets for Image Copies at the Local Site

Overview

This unit contains information on defining datasets for image copies.

Purpose

The image copy datasets become the copy files for the primary and backup copies at the local site.

Access

Select option 9 on the Utility Profile menu to directly access the Copy Options panel for copy files. Access this panel in sequence by pressing Enter on the Copy Utility Options panel.

Elements of the Copy Options panel

This is the Copy Options panel for the local site.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  ----
Cmd ===>                                                    yy/mm/dd

                     C O P Y    O P T I O N S

 COPY Files (for the primary and backup copy at the LOCAL site)
    1 DDname1 ===>             # = COPYnnnn, blank = "spacenm", default =
      2 UNIT1 ===> SYSDA       Device type for Copy file 1.
      3 SIZE1 ===>             Override Copy dataset size (tracks).
   4 STORCLAS ===>               5 DATACLAS===>              6 MGMTCLAS===>
      7 STACK ===> N           All datasets on one tape? Y/N

    1 DDname2 ===>             # = COPYnnnn 1
      2 UNIT2 ===> SYSDA       Device type for Copy file 2.
  4. STORCLAS ===>               5 DATACLAS===>              6 MGMTCLAS===>
      7 STACK ===> N           All datasets on one tape? Y/N
```

Elements of the Copy Options panel (continued)

1 DDnames of the primary and backup datasets at the local site:

- To use the DDname from the utility skeleton JCL, type **SYSCOPY**.
- To use the table space or index name as the DDname, leave the field blank.
- To generate names sequentially in the form COPY0001, COPY0002, and so forth, type **#**.

2 Name of the device types for the primary and backup data set work files on both the local and recovery sites. The default is SYSDA.

3 Primary space allocation for the primary local dataset:

- If you leave this field blank, !DB/SMU computes the required size and eliminates the problem of correctly estimating it.
- If you specify the size, then no size calculations are performed.

4 Storage attributes for the datasets

5 Data class for a new dataset

6 Management class for an SMS-managed dataset

7 Storage option for copied datasets; you *must* enter a value in this field:

Y Store all the datasets on one tape
N Do not store all the datasets on one tape

Datasets for Image Copies at the Recovery Site

Overview

This unit contains information on defining datasets for image copies.

Purpose

These image copy datasets become the recovery files for the primary and backup copies at the recovery site.

Access

Select option 10 on the Utility Profile menu to directly access the Copy Options panel for copy files at the recovery site. Access this panel in sequence by pressing Enter on the Copy Options panel for the local site.

Elements of the Copy Options panel

This is an example of the Copy Options panel for the recovery site.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  ----
Cmd ===>                                                    mm/dd/yy

                     C O P Y    O P T I O N S

 COPY Files (for the primary and backup copy at the RECOVERY site)
    1 DDname1 ===>             # = RECOnnnn
      2 UNIT1 ===> SYSDA       Device type for Recovey Dataset 1.
    3 STORCLAS ===>              4 DATACLAS===>              5 MGMTCLAS===>
      6 STACK ===> N           All datasets on one tape? Y/N

    1 DDname2 ===>             # = RECOnnnn 1
      2 UNIT2 ===> SYSDA       Device type for Recovery Dataset 2.
    3 STORCLAS ===>              4 DATACLAS===>              5 MGMTCLAS===>
      6 STACK ===> N           All datasets on one tape? Y/N
```

1 DDnames of the primary and backup datasets respectively. To gnerate names sequentially in the form RECO0001, RECO0002, and so forth, type **#**.

2 Name of the device types for the primary and backup files respectively. The default is SYSDA.

3 Storage attributes for the utility datasets

Elements of the Copy Options panel (continued)

4 Data class for a new dataset

5 Management class for an SMS-managed dataset

6 Storage option for copied datasets; you *must* enter a value in this field:

Y Store all the datasets on one tape
N Do not store all the datasets on one tape

Options for the MERGECOPY Utility

Overview

This unit contains information on defining options for the MERGECOPY utility.

Purpose

These options specify conditions for merging 1 or more incremental copies of a table space into a single incremental copy or into a new full-image copy.

Access

Select option 11 on the Utility Profile menu to directly access the MERGECOPY Utility Options panel. Access this panel in sequence by pressing Enter on the Copy Options panel for the recovery site.

Elements of the Mergecopy Utility panel

This is the Mergecopy Utility panel.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  ----------------------
CMD ===>                                                      YY/MM/DD 10:48

             M E R G E C O P Y   U T I L I T Y   O P T I O N S

  1 UTILITY-ID ===>               UNIQUE UTILITY ID, DFLT IS USERID.JOBNAME.

       2 START ===> P          LEAVE BLANK IF NOT RESTARTING THIS UTILITY.
                               ELSE, RESTART AT C COMMIT, OR P PHASE.

 3 CMDS BEFORE ===> P          P -STOP  R -ST RO  U -ST UT  V = P&U.
  4 CMDS AFTER ===> W          P -STOP  R -ST RO  W -ST RW  Z = PRTA

    5 NEWCOPY ===> N           Y MERGE WITH FULL COPY TO FORM NEW FULL COPY.
                               N MERGE ONLY INCREMENTAL COPIES.

   6 COPYDDNM ===> PRINT       # = COPYNNNN, BLANK = SPACENM, DEFAULT = SYSCOPY.
   7 COPYUNIT ===> SYSDA       DEVICE TYPE FOR COPY OUTPUT DATASETS.
   8 COPYSIZE ===> 104         OVERRIDE COPY OUTPUT DATASET ALLOCATION SIZE.

   9 WORKDDNM ===> PRINT       # = WORKNNNN, BLANK = SPACENM, DEFAULT = SYSUT1.
  10 WORKUNIT ===> SYSDA       DEVICE TYPE FOR WORK FILE.  NO IF NONE.
  11 WORKSIZE ===> 101         OVERRIDE WORK DATASET ALLOCATION VALUE.
```

Elements of the Mergecopy Utility panel (continued)

1 Unique identifier necessary for each DB2 utility. This option sets the value of UID in the JCL EXEC statement.

2 Restart option:

P Restarts the MERGECOPY utility at the beginning of a phase or at the last commit state. This option sets the value of UTPROC in the JCL EXEC statement.

blank Does not restart the utility.

3 DB2 commands option; see "Including DB2 Commands in Your Utility Job Stream" on page 398.

4 DB2 commands option; see "Including DB2 Commands in Your Utility Job Stream" on page 398.

5 Output format:

Y Creates a new full image copy
N Creates only incremental image copies

6 DDname of the copy dataset that becomes the DDname for *every* occurrence of the dataset. (This can result in errors for COPYDDNM if you are running the utility for more than one table space. To avoid errors, *do not* speciy ONE STEP=Y.)

- To use the name from the skeleton JCL, type **SYSCOPY**.
- To use the table space name as the COPYDDNM name, leave the field blank.
- To have the utility generate DDnames sequentially as COPY0001, COPY0002, and so forth, type **#**.

7 Device type used for the UNIT and DEVT keywords in the JCL statements

8 Primary space allocation for the dataset:

- If you leave this field blank, !DB/SMU computes the required size and eliminates the problem of correctly estimating it.
- If you specify the size, then no size calculations are performed.

Elements of the Mergecopy Utility panel (continued)

9 DDname in the JCL DD statement for the work dataset:

- To use the names from the skeleton JCL, type **SYSUT1**. This is the recommended choice for the WORKDDNM.
- To use the table space space name as the DDname, leave the field blank.
- To have the utility generate DDnames sequentially for WORK0001, WORK0002, and so forth, type **#**.

If you type **#** or leave the field blank and use the utility skeleton JCL, modify the JCL repeating section to use the correct DDnames.

10 Device type for the work dataset. The default value is SYSDA. Type the name of the device type for the work files, or type **NO** for none.

11 Primary space allocation for the dataset:

- If you leave this field blank, !DB/SMU computes the required size and eliminates the problem of correctly estimating it.
- If you specify the size, then no size calculations are performed, and the utility generation will run faster.

Options for the MODIFY Utility

Overview

This unit contains information on defining options for the MODIFY utility.

Purpose

These options enable you to delete unwanted image copy records from the DB2 SYSCOPY table and the related log records from the DB2 log.

Access

Select option 12 on the Utility Profile menu to directly access the Modify Utility Options panel. Access this panel in sequence by pressing Enter on the Mergecopy Utility Options panel.

Elements of the Modify Utility Options panel

This is the Modify Utility Options panel.

```
UTILITY PROFILES ---------------  DB/SMU  ---  DB2=D51Y  -----------------------
Cmd ===>                                                          yy/mm/dd 10:52

                  MODIFY UTILITY OPTIONS

1 UTILITY-ID  ===>                 Must be unique for each Utility.
                              The default is 'Userid.Jobname'.

2 START       ===>            Leave blank if not restarting this Utility.
                              Else, restart at C Commit, or P Phase.

3 AGE         ===>            Delete all records older than AGE days. (0 = *)
4 DATE        ===> 19850101   Delete all records older than DATE.
                   YYYYMMDD   Be careful with the format of the date.
```

Elements of the Modify Utility Options panel (continued)

1 Utility identifier. This value must be unique. It sets the value of UID in the JCL EXEC statement.

2 Restart option for the utility. This option sets the UTPROC value in the JCL EXEC statement.

C Restarts at the last commit point
P Restarts at the beginning of the last executed phase
blank Does not restart the MODIFY utility

3 Option to delete all records older than the specified number of days. The range is 1–32,767.

4 Option to delete all records written before the date specified. Use the format *yymmdd.*

The date field *must* contain a value.

Options for the QUIESCE Utility

Overview

This unit contains information on defining options for the QUIESCE utility.

Purpose

These options enable you to stop further changes from being made to the database and to record the relative byte address (RBA) of the quiesce point.

Access

Select option 13 on the Utility Profile menu to directly access the Quiesce Utility Options panel. Access this panel in sequence by pressing Enter on the Modify Utility Options panel.

Elements of the Quiesce Utility Options panel

This is the Quiesce Utility Options panel.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  ----------------------
Cmd ===>                                                      yy/mm/dd 12:48

                Q U I E S C E   U T I L I T Y   O P T I O N S

 1 CMDS BEFORE ===> P        P -Stop, R -St RO, U -St UT, V = P&U.

  2 CMDS AFTER ===> R        P -Stop, R -St RO, W -St RW, Z = PRTA


 3 WRITE OPTION ===> Y       Y -Write(default) modified bufferpool pages.
                             N -Do not write modified bufferpool pages.
```

Elements of the Quiesce Utility Options panel (continued)

1 DB2 commands option; see "Including DB2 Commands in Your Utility Job Stream" on page 398.

2 DB2 commands option; see "Including DB2 Commands in Your Utility Job Stream" on page 398.

3 Option to write modified bufferpool pages to the space.

Y Write out modified pages
N Do not write out modified pages

Options for Recovering Table Spaces

Overview

This unit contains information on defining options for the RECOVERY utility.

Purpose

These options apply to recovering DB2 data from image copies of a table space and, if required, subsequent table space change log records.

Access

Select option 14 on the Utility Profile menu to directly access the first Recovery Utility Options panel. Access this panel in sequence by pressing Enter on the Quiesce Utility Options panel.

Elements of the first Recovery Utility Options panel

This is the first of two Recovery Utility Options panels.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  ----------------------
Cmd ===>                                                      yy/mm/dd 10:48

          R E C O V E R Y   U T I L I T Y   O P T I O N S   (1 of 2)

 1 CMDS BEFORE ===> P        P -Stop, R -Start RO, U -Start UT, V = P&U.

  2 CMDS AFTER ===> R        P -Stop, R -Start RO, W -Start RW, Z = PRTA

 3 RECOVER SET? ===>         Y to Recover all listed Table Spaces as a set.


  If Recovering a single page, specify PAGE and, optionally, CONT.

        4 PAGE ===>          Number of page to be recovered (decimal digits,
                             or hex digits between quotes preceded by "X").

        5 CONT ===> N        Y to continue if a previous Recovery failed.

   Press ENTER for more Recovery Options.
```

Elements of the first Recovery Utility Options panel (continued)

1 DB2 commands option; see "Including DB2 Commands in Your Utility Job Stream" on page 398.

2 DB2 commands option; see "Including DB2 Commands in Your Utility Job Stream" on page 398.

3 Option to recover all the table spaces together to the same quiesce point. Type **Y** to process all the selected table spaces as a set.

This option requires DB2 Version 2.0 or higher and an established quiesce point for the set. !DB/SMU ignores this option if you issued the UTIL command from a !DB®/WORKBENCH Set panel because the table spaces are already grouped into sets.

4 Particular page of the space to be recovered. To recover a single page, type the page number in either decimal digits or hexadecimal digits. If you are using hexadecimal digits, the page number must be typed between single quotes and preceded by X.

5 Option to continue recovery after a page has been repaired:

Y Indicates recovery is to continue
N Indicates recovery is not to continue

Options for Controlling the Scope of a Table Space Recovery

Overview

This unit contains information on the options that control the scope of a recovery for table spaces.

Purpose

The settings that control the scope of a recovery for table spaces are optional. You can choose an option to limit the scope of the recovery, or you can bypass these options if you choose. To perform a full recovery, leave all options blank or enter the value needed to bypass the option.

Access

Select option 15 on the Utility Profile menu to directly access the second Recovery Utility Options panel. Access this panel in sequence by pressing Enter on the first Recovery Utility Options panel.

Elements of the second Recovery Utility Options panel

This is an example of page 2 of the Recovery Utility Options panel

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  ----------------------
Cmd ===>                                                      yy/mm/dd 10:48


          R E C O V E R Y   U T I L I T Y   O P T I O N S   (2 of 2)

 These options limit the recovery scope.  Fill in values and press ENTER.

 Choose none or one of:

1 ERR-RANGE ===>              Y to recover pages with reported I/O errors.

2 or  TORBA ===>              Hex string (Log Relative Byte) to stop recovery.

3 or TOCOPY ===> 'TDDB.DDH.TEST.DATASET'                  Copy DS Name.
                 This names a specific Image Copy dataset.  Enclose the
                 name in quotes if it is not prefixed with your Userid.

 If the TOCOPY name is not unique in table SYSIBM.SYSCOPY, identify further.

    4 CATALOGED? ===> N       Y if cataloged, N if not.

 5 or      VOLSER ===>        Volume Serial number.
 6 and        SEQ ===>        File sequence number on the volume.
```

Elements of the second Recovery Utility Options panel (continued)

1 Option to recover pages that fall within the range of pages you specify and have reported I/O errors:

Y Recover all pages within the range having reported I/O errors
N Do not recover the pages having reported I/O errors

2 Byte string greater than the RBA of the last log record to be included in the recovery. Leave this field blank to bypass this option.

3 Specific image copy data set from which to recover. To use a specific dataset, type its name in this field. Leave this field blank to bypass this option.

If the dataset you specify is a full image copy, !DB/SMU uses only this copy to perform the recovery. If the dataset is an incremental copy, !DB/SMU also uses the previous full image copy and intervening incremental copies.

Complete the following fields for the TOCOPY dataset if more than one image copy dataset has same name:

CATALOGED
Indicates whether or not the dataset you specified is cataloged:

Y dataset is cataloged
N dataset is not cataloged

VOLSER
If the dataset is not cataloged, enter the volume serial number of the dataset in this field.

SEQ If the file sequence number of the image copy dataset is needed to further identify the file on a tape, type it here.

Options for Reorganizing Table Spaces

Overview

This unit contains information on defining options for the REORG utility.

Purpose

These options define operations for reoganizing a table space.

Access

Select option 16 on the Utility Profile menu to directly access the first Reorg Utility Options panel. Access this panel in sequence by pressing Enter on the second Recovery Utility Options panel.

Elements of the first Reorg Utility Options panel

This panel is the first of two Reorg Utility Options panels.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  -------------
Cmd ===>                                                      yy/mm/dd

                 R E O R G   U T I L I T Y   O P T I O N S

  1 CMDS BEFORE ===>           P -Stop, R -Start RO, U -Start UT, V = P&

  2 CMDS AFTER ===>            P -Stop, R -Start RO, W -Start RW, Z = P&

  3 AFTER UNLD ===> C          C (CONTINUE), P (PAUSE) or O (UNLOAD ONLY

  4 SORT DATA  ===>            Y/N  Y to do a TS scan and then External

  5 RELOAD LOG ===> N          Y/N Log during Reload?  N is faster.

  6 COPY AFTER ===> Y          Y/N Generate Full Image Copy after Reorg.

  7 RUNSTATS   ===>            Y -Perform RUNSTATS/REBIND After, N -No

 8 KEEPDICTIONARY ===>         Y/N to reuse Compression Dictionary
 9 SORTDEVT ===> SYSDA         The Device Type used for Sort Work files.
10 SORTNUM ===>                The number of data sets to be dynamically
                               allocated by the Sort program.
```

1 DB2 commands option; see “Including DB2 Commands in Your Utility Job Stream” on page 398.

2 DB2 commands option; see “Including DB2 Commands in Your Utility Job Stream” on page 398.

Elements of the first Reorg Utility Options panel (continued)

3 Action to take after the data has been unloaded:

C Continue
P Pause
O Stop

4 Whether to perform an external sort of the data:

Y Perform an external sort
N Do not perform an external sort

5 Whether to log records during the reload phase:

Y Log records during the reload phase
N Do not log records during the reload phase

6 Whether to execute a full image copy after the REORG has completed:

Y Execute a full image copy
N Do not execute a full image copy

If you specify **Y**, you should ensure that you do not inadvertently include an additional copy in your job stream. Doing so will cause !DB/SMU to image copy your space twice.

7 Whether to generate statistics and rebind plans and packages after the REORG has completed:

Y Generate statistics and perform a rebind
N Do not generate statistics or perform a rebind

8 Whether to reuse the current compression dictionary:

Y Reuse the current compression dictionary
N Do not reuse the current compression dictionary

9 Name of the device type for the dynamically allocated SORT datasets. The default value is SYSDA. (If you omit a value in SORTNUM, the system ignores any text in this field.)

10 Number of datasets to be allocated dynamically by SORT. (If you type a number in this field, you must type a device type in SORTDEVT.)

Datasets for Reorganizing Table Spaces

Overview

This unit contains information on defining datasets for the reorganizing a table space.

Purpose

These options specify dataset name, size, and class information.

Access

Select option 17 on the Utility Profile menu to directly access the second Reorg Utility Options panel. Access this panel in sequence by pressing Enter on the first Reorg Utility Options panel.

Elements of the second Reorg Utility Options panel

This is an example of the second Reorg Utility Options panel.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  ----
Cmd ===>                                                yy/mm/dd 10:54


                  R E O R G    U T I L I T Y    O P T I O N S

 WORK Files
     1 DDname ===> #           # = WORKnnnn, blank = "spacenm", default = SYSUT1.
       2 UNIT ===> SYSDA       Device type for Work files.
       3 SIZE ===> 101         Override Work dataset allocation value.
    4 STORCLAS===> S1            5 DATACLAS===> D1           6 MGMTCLAS===> M1

 UNLOAD Files
     1 DDname ===> #           # = UNLDnnnn, blank = "spacenm", default = SYSREC.
       2 UNIT ===> SYSDA       Device type for Unload datasets.
       3 SIZE ===> 102         Override Unload dataset allocation value.
    4 STORCLAS===> S2            5 DATACLAS===> D2           6 MGMTCLAS===> M2

 SORTOUT Files
     1 DDname ===> #           # = SOUTnnnn, blank = "spacenm", default = SORTOUT.
       2 UNIT ===> SYSDA       Device type for SORTOUT files.
       3 SIZE ===> 103         Override SORTOUT dataset allocation value.
    4 STORCLAS===> S3            5 DATACLAS===> D3           6 MGMTCLAS===> M3
```

Elements of the second Reorg Utility Options panel (continued)

1 Names of the WORK, UNLOAD, and SORTOUT datasets:

- To use the name from the skeleton JCL for the work dataset, type **SYSUT1**. This option is recommended for the work DDname.
- To use the name from the skeleton JCL for the unload dataset, type **SYSREC**. This option can result in errors for the unload DDname if you run the utility for more than one table space and specify **ONE STEP=Y**.)
- To use the name from the skeleton JCL for the sort output dataset, type **SORTOUT**.
- To use the table space or index space name as the DDname, leave the field blank.
- To generate DDnames sequentially as WORK0001, UNLD0001, or SOUT0001, type **#**. (This option is recommended for UNLDDDNM and SOUTDDNM.)

If you use the utility skeleton JCL and type **#** for the DDname or leave the field blank, you must modify the JCL repeating section to use the correct DDnames.

2 Name of the device type of the dataset. The default is SYSDA.

3 Primary space allocation for the dataset:

- If you leave this field blank, !DB/SMU computes the required size and eliminates the problem of correctly estimating it.
- If you specify the size, then no size calculations are performed.

4 Storage attributes for the utility datasets

5 Data class for a new dataset

6 Management class for an SMS-managed dataset

RUNSTATS Options for Table Spaces

Overview

This unit contains information on defining table space options for the RUNSTATS utility.

Purpose

Use this utility to gather information about table space utilization and efficiency and store the information in the DB2 catalog.

Access

Select option 18 on the Utility Profile menu to directly access the RUNSTATS Utility Options panel for table spaces. Access this panel in sequence by pressing Enter on the second Reorg Utility Options panel.

Elements of the RUNSTATS Utility Options panel

This is the RUNSTATS Utility Options panel.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D51Y  ----------------------
Cmd ===>                                                        yy/mm/dd 10:54

                        RUNSTATS UTILITY OPTIONS
1 CMDS BEFORE ===> V    P = Stop, R = Start RO, U = Start UT, or V = P and U

2 CMDS AFTER  ===> Z    P = Stop, R = Start RO, W = Start RW, or Z = P and R

3 SHRLEVEL    ===> R    R = REFERENCE (allow only read processing) or
                        C = CHANGE (allow updates) while RUNSSTATS runs

4 INDEX       ===> *    * = all indexes or blank = none
    KEYCARD   ===> Y    Collect cardinalities for each key column? Y or N
    FREQVAL   ===> Y    Collect frequent value statistics?  Y or N
      NUMCOLS ===> 5    number of key columns to concatenate
      COUNT   ===> 6    number of frequent values to collect

5 REPORT      ===> Y    Print report?  Y or N

6 UPDATE      ===> N    Update the Catalog?
                        A = all, P = Accesspath, S = Space, or N = none

7 TABLE STATS ===> N    Gather stats for non-indexed columns?  Y or N
```

Elements of the RUNSTATS Utility Options panel (continued)

1 DB2 commands option; see "Including DB2 Commands in Your Utility Job Stream" on page 398.

2 DB2 commands option; see "Including DB2 Commands in Your Utility Job Stream" on page 398.

3 Share access for the space RUNSTATS Utility is executing:

R Allow reference only to the space
C Allow reference, change, or update to the space

4 Option to gather RUNSTATS statistics for indexes associated with the table space:

***** Gather statistics for all indexes associated with the table space
blank Do not gather statistics
KEYCARD
Activates the collection of cardinalities for each key column.
FREQVAL
Activates the collection of frequent value statistics.
NUMCOLS
This option is inactive unless the FREQVAL option is active. Determines the number of key columns to concatenate.
COUNT
This option is inactive unless the FREQVAL option is active. Determines the number of frequent values to collect.

5 Whether or not to produce a report:

Y Produce a report
N Do not produce a report

6 Whether to update the catalog:

A Update all catalog information
P Update only the access path information
S Update only the table space information
N Do not update the catalog

7 Whether to gather statistics for nonindexed columns (only applicable if you are running DB2 Version 2 and later):

Y Gather statistics for nonindexed columns
N Do not gather statistics for nonindexed columns

Chapter 23. Options for Creating DB2 Utility Profiles for Index Spaces and BIND/REBIND Plans and Packages

Introduction

This chapter and the two preceeding contain information about creating profiles for the DB2 utilities. This chapter discusses the panels specific to index spaces as well as the BIND/REBIND utility. The preceding chapters discussed the use of the general options panels and the panels that apply specifically to table spaces, respectively.

Chapter Contents

Preparing to Create Utility Profiles

Overview

This unit provides information about using a series of panels to create a utility profile required to run a DB2 utility.

Background about creating a utility profile

!DB/SMU gives you the ability to execute specific DB2 utilities during batch and online operations. The execution of a utility requires information you specify by means of a utility profile. You can create a utility profile tailored to your needs by using a series of panels that you access from the Utility Profiles panel.

Organization of information to create a utility profile

!DB/SMU provides a series of 22 panels with which you can interact to specify values for utility profile options. The information you need to use these panels is structured as shown in this chart:

Task You Want to Perform	Source of Information
Provide values for general options that apply to both table spaces and index spaces.	"General Options for Creating DB2 Utility Profiles" on page 393
Provide values for options that apply to table spaces	"Options for Creating DB2 Utility Profiles for Table Spaces" on page 415
Provide values for options that apply to index spaces	This chapter
Review additional detailed information about all of the panels and their fields	Online help for each panel
Review a detailed description of the DB2 utilities and their options	*IBM DATABASE 2 Command and Utility Reference*

Options for Recovering Indexes

Overview

This unit contains information on defining index recovery options for the RECOVER utility.

Purpose

Use this utility to rebuild the DB2 indexes from the table spaces that contain the indexed tables.

Access

Select option 19 on the Utility Profile menu to directly access the Recover Index panel. Access this panel in sequence by pressing Enter on the RUNSTATS Utility Options panel.

Elements of the Recover Utility Options panel for indexes

This is the Recover Index Utility Options panel.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  ----------------------
Cmd ===> right 20                                              yy/mm/dd 10:54

          R E C O V E R   I N D E X   U T I L I T Y   O P T I O N S

 1 CMDS BEFORE ===>           P -Stop, R -Start RO, U -Start UT, V = P&U.

 2 CMDS AFTER  ===>           P -Stop, R -Start RO, W -Start RW, Z = PRTA
```

1 DB2 commands option; see "Including DB2 Commands in Your Utility Job Stream" on page 398.

2 DB2 commands option; see "Including DB2 Commands in Your Utility Job Stream" on page 398.

Options for Reorganizing Index Spaces

Overview

This unit contains information on defining index recovery options for the RECOVER utility.

Purpose

Use this utility to rebuild the DB2 indexes from the table spaces that contain the indexed tables.

Access

Select option 20 on the Utility Profile menu to directly access the first REORG Index Utility Options panel. Access this panel in sequence by pressing Enter on the Recover Index Utility Options panel.

Elements of the first Reorg Index Utility Options panel

This is an example of the first Reorg Index Utility Options panel.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  ----------------------
Cmd ===>                                                    yy/mm/dd 10:54

           R E O R G   I N D E X   U T I L I T Y   O P T I O N S

 1 UTILITY-ID ===>            Unique Utility ID, default is Userid.Jobname.

      2 START ===>            Leave blank if not restarting this Utility.

 3 CMDS BEFORE ===>           P -Stop, R -Start RO, U -Start UT, V = P&U.

 4 CMDS AFTER ===>            P -Stop, R -Start RO, W -Start RW, Z = PRTA

 5 AFTER UNLD ===> C          C (CONTINUE), P (PAUSE) or O (UNLOAD ONLY).
```

Elements of the first Reorg Index Utility Options panel (continued)

1 Specifies a unique utility ID and sets the value of the UID variable in the JCL EXEC statement.

2 Restart option:

C Restart at the beginning of a phase or at the last commit point
blank Do not restart this utility

3 DB2 commands option; see "Including DB2 Commands in Your Utility Job Stream" on page 398.

4 DB2 commands option; see "Including DB2 Commands in Your Utility Job Stream" on page 398.

5 Continue option:

C Continue after data has been unloaded
P Pause after data has been unloaded
O Unload only

Datasets for Reorganizing Index Spaces

Overview

This unit contains information on defining datasets for reorganizing an index space.

Purpose

These options specify dataset name, size, and class information.

Access

Select option 21 on the Utility Profile menu to directly access the second REORG Index Utility Options panel. Access this panel in sequence by pressing Enter on the first Reorg Index Utility Options panel.

Elements of the second Reorg Index Utility Options panel

This is an example of the second Reorg Index Utility Options panel.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D31A  ----
Cmd ===>                                                99/02/08 10:54


           R E O R G   I N D E X   U T I L I T Y   O P T I O N S

 UNLOAD Files
   1 DDname ===> #          # = UNLDnnnn, blank = "spacenm", default = SYSREC.
     2 UNIT ===> SYSDA      Device type for Unload datasets.
     3 SIZE ===> 102        Override Unload dataset allocation size.


 WORK Files
   1 DDname ===> #          # = WORKnnnn, blank = "spacenm", default = SYSUT1.
     2 UNIT ===> SYSDA      Device type for Work files.
     3 SIZE ===> 101        Override Work dataset allocation value.


 SORTOUT Files
   1 DDname ===> #          # = SOUTnnnn, blank = "spacenm", default = SORTOUT.
     2 UNIT ===> SYSDA      Device type for SORTOUT files.
     3 SIZE ===> 103        Override SORTOUT dataset allocation value.
```

Elements of the second Reorg Index Utility Options panel (continued)

1 Data definition name you want assigned in the JCL DD statement:

- To use the names from the skeleton JCL, type **SYSREC**, **SYSUT1**, and **SORTOUT**;
- To use the table space or index space name DDname, leave the field blank;
- To generate sequential DDnames (such as WORK0001, UNLD0001, or SOUT0001), type **#**.

If you type **#** or leave the field blank and use the utility skeleton JCL, modify the JCL repeating section to use the correct DDnames.

2 Device type you want to use for the dataset. The default is SYSDA.

3 Dataset allocation size in tracks:

- If you leave this field blank, !DB/SMU computes the required size and eliminates the problem of correctly estimating it.
- If you specify the size, then no size calculations are performed.

RUNSTATS Options for Index Spaces

Overview

This unit contains information on defining index space options for the RUNSTATS utility.

Purpose

Use this utility to gather information about index space utilization and efficiency, and store the information in the DB2 catalog.

Access

Select option 22 on the Utility Profile menu to directly access the RUNSTATS Index Utility Options panel. Access this panel in sequence by pressing Enter on the second Reorg Index Utility Options panel.

Elements of the RUNSTATS Index Utility Options panel

This is an example of the RUNSTATS Utility Options panel for indexes.

```
UTILITY PROFILES ---------------   DB/SMU  ---  DB2=D51Y  ----------------------
Cmd ===>                                                        yy/mm/dd 10:54


                             RUNSTATS INDEX UTILITY OPTIONS

 1 CMDS BEFORE ===> V        P = Stop, R = Start RO, U = Start  UT, V = P and U

 2 CMDS AFTER  ===> Z        P = Stop, R = Start RO, W = Start  RW, Z = P and R

 3 SHRLEVEL    ===> R        R  = REFERENCE (allow only read processing) or
                             C  = CHANGE (allow updates) while RUNSTATS runs

 4 REPORT      ===> Y        Report the values? Y/N

 5 UPDATE      ===> N        Update the Catalog? A = All, P =  Accesspath

 6 KEYCARD     ===> Y        Collect cardinalities for each key column? Y/N

   FREQVAL     ===> Y        Collect frequent value statistics?       Y/N
     NUMCOLS   ===> 5        number of key columns to concatenate
     COUNT     ===> 6        number of frequent values to collect
```

Elements of the RUNSTATS Index Utility Options panel (continued)

1 DB2 commands option; see "Including DB2 Commands in Your Utility Job Stream" on page 398.

2 DB2 commands option; see "Including DB2 Commands in Your Utility Job Stream" on page 398.

3 Share access for the space against which the RUNSTATS Utility is executing:

R Allows reference only to the space
C Allows reference, change, or update to the space

4 Report request:

Y Produce a report
N Do not produce a report

5 Whether to update the catalog:

A Update all catalogs
P Update access path information only

6 Option to gather RUNSTATS statistics for indexes associated with the table space:

Y Gather statistics
N Do not gather statistics

BIND/REBIND Options for Plans and Packages

Overview

This unit contains information for enabling or disabling BIND/REBIND utility options.

Purpose

Use this utility to set the BIND/REBIND utility options for both Plans and Packages.

Access

Select option 23 on the Utility Profile menu to directly access the BIND/REBIND Utility Options panel. Access this panel in sequence by pressing Enter on the RUNSTATS Index Utility Options panel.

Elements of the BIND/REBIND Utility Options panel

This is an example of the BIND/REBIND Utility Options panel for Plans and Packages.

```
UTILITY PROFILES ---------------  DB/SMU  ---  DB2=D51Y  --------------
Cmd ===>                                                      99/09/30

                      BIND/REBIND UTILITY OPTIONS

 1 PREOPT    ===> Y   Y = REOPT(VARS)              |
                      N = NOREOPT(VARS)            |
                                                   |== PLAN options
 2 PDEFER    ===> Y   Y = DEFER(PREPARE)           |
                      N = NODEFER(PERPARE)         |

 3 KREOPT    ===> Y   Y = REOPT(VARS)              |
                      N = NOREOPT(VARS)            |
                                                   |== PACKAGE options
 4 KDEFER    ===> Y   Y = DEFER(PREPARE)           |
                      N = NODEFER(PERPARE)         |

Notes:
  1. Specifying Y for PREOPT, implies a value of Y for PDEFER.
  2. If the value of KREOPT is Y, KDEFER defaults to Y.
  3. If the value of KREOPT is N, KDEFER defaults to the value of PREOPT for a
     local BIND/REBIND or to N for a remote BIND/REBIND.
```

Elements of the BIND/REBIND Utility Options panel (continued)

1 Plan REOPT request:

Y Enable Plan REOPT
N Disable Plan REOPT

2 Plan DEFER request:

Y Enable Plan DEFER
N Disable Plan DEFER

3 Package REOPT request:

Y Enable Package REOPT
N Disable Package REOPT

4 Package DEFER request:

Y Enable Package DEFER
N Disable Package DEFER

Batch Operations

Chapter 24.
Introduction to Batch Operations for !DB/SMU

Introduction

You can perform most !DB/SMU online operations in batch. This includes running scans and space maps, initiating DB2 utilities and printing reports. You can use batch processing to automate many of the tasks involved in maintaining DB2 and optimizing data performance. This chapter introduces the batch operations for !DB/SMU

Chapter Contents

Prerequisites for the Using !DB/SMU Batch Facility

Overview

This unit describes the prerequisites you must accomplish before executing a batch job. It also addresses some considerations for moving from release to release of !DB/SMU.

Prerequisites for batch jobs

Do the following before submitting a batch job:

- Review the user options specified on the administrative panels for housekeeping to confirm that they meet your needs.
- Because the flow of batch operations closely follows the flow of online operations, you should be familiar with the online operations and reports for the scan, space map, and monitor functions.

Existing batch jobs

Recreate any existing jobs, including those in a job scheduler, that were created with Version 220 or earlier of !DB/SMU. Such jobs can contain statements that are inappropriate for the current version of !DB/SMU.

Background about Batch Processing

Overview

This unit provides background about using batch operations with !DB/SMU

Authorization for batch jobs

You must have OPTS authorization to delete or modify a profile that was created by someone else.

Control of batch jobs

!DB/SMU provides you with full control of your batch jobs. After a batch job has been created, you can take any of the actions listed below.

Actions you can take against a batch job

!DB/SMU automatically generates the job stream for your batch job and saves it in the user PDS. After you generate a job stream, you can do any of the following:

- Submit the job stream
- Edit the job stream
- Delete the job stream
- Print the job stream
- Copy the job stream to the log PDS

Extract considerations for batch jobs

The monitor and space map run profiles and the exception profiles use the information in the extract to generate reports and initiate actions. If you have created or changed a database since the !DB/SMU extract was last run, you will need to rerun the extract to ensure that those changes are reflected. For example, any newly created table spaces would not be listed in the reports or affected by !DB/SMU actions, such as performing an incremental copy.

Online help for creating batch jobs

Online help provides additional information for using the batch interface to create profiles and generate batch jobs.

Performance considerations for batch jobs

!DB/SMU is designed to achieve optimum speed and efficiency. When scanning many table spaces (for example, all of the table spaces in a given subsystem), it is more efficient to break a single batch job into multiple steps or jobs. You can also consider running parallel jobs (see "Building and Executing Batch Jobs in Parallel" on page 484). These techniques require that you create a few more run profiles, but generally result in savings of both CPU and elapsed time.

Batch output

!DB/SMU provides various types of output reports that provide the following information:

- Comprehensive scan statistics for table spaces and index spaces
- DASD managment and usage information
- Dumps for selected pages
- Data integrity problems
- Exception conditions that do not meet criteria you estabish, with a description of the value that generated the exception and the resulting action !DB/SMU has taken
- Formatted information for selected table space pages or index pages
- Results of monitoring spaces for DB2 dataset size, copy factor, reorg factor, and number of extents
- Summary information derived from space map pages about page quantities, page use, DASD extents, the copy factor, and the reorg factor

See "Using the Batch Reports for !DB/SMU" on page 591 for detailed information and examples of the batch reports.

Return codes for batch jobs

The return codes for batch jobs provide you with information about the jobs. Each batch job returns one of these codes. !DB/SMU returns the code that represents the most serious error condition found.

Return Code	Meaning
0	No exception conditions were found.
2	An informational message that exception conditions were found.
4	A warning that exception conditions were found, but were not serious enough to cause !DB/SMU to terminate the job. Refer to the messages !DB/SMU issued to determine the extent of the problem and potential avenues for correction.
8	Exception conditions were found that caused significant loss of function. !DB/SMU terminated the job.

Benefits of Automating Activities Using Batch Operations

Overview

The !DB/SMU batch facility lets you automate DB2 management tasks. This unit explains the advantages that automation offers.

Summary of benefits

Automating DB2 activities by using the batch facility provides these benefits:

- You can schedule and submit !DB/SMU batch jobs at your convenience and run them without having to monitor their progress
- You can be proactive in tuning your DB2 system for improved performance. !DB/SMU assists you in spotting trends and fixing hot spots
- You can reduce errors when creating job streams, because !DB/SMU automatically creates much of the required code for you.
- You control the execution of a batch job. You always have the option to edit a job before it is submitted, submit the job as !DB/SMU created it, save the job for execution at a time you choose, or discard the job.
- You can spend less time performing routine tasks by using the !DB/SMU monitoring facilities to trigger automated system activities.

Common activites you can automate in batch jobs

The activities that follow are among those you can automate using !DB/SMU batch processing:

- Monitor space management and performance; report performance information in comprehensive batch reports
- Identify data integrity errors; identify and act on conditions that fail to meet criteria you specify
- Initiate DB2 and OEM utilities

Examples of monitoring space management and performance

This chart provides examples of the use of batch jobs to monitor space management and data performance.

Automated Activity	Benefit
Monitor DASD usage	Reports the number and location of any extents for DB2 datasets
Scan selected space map pages	Reports summary information on the number of pages, number of extents, percentage of used pages, and the copy and reorg factors
Scan selected table spaces and index spaces	Prints detailed statistical reports and produces a variety of usage graphs for analysis purposes
Determine whether you have the overhead of compression without real savings.	Scan reports provide the statistics for data compression, free space, and DASD usage.

Examples of identifying integrity errors or exception conditions

This chart provides examples of the use of batch jobs to identify integrity errors. It also provides examples of the identification of exception conditions and actions that you can take.

Automated Activity	Benefit
Check DB2 for data integrity errors	Reports and saves information about errors, which you can correct later with the repair facility
Monitor DB2 for exception conditions	Finds exceeded limits and initiates an appropriate action
Call a user-written exit	Generates comprehensive instructions for monitoring exception conditions or performs any instructions programmed in the exit

Examples of automating space management or initiating DB2 or OEM utilities

This chart provides examples of instances where you might want to build a complex batch job or initiate appropriate DB2 utilities.

Automated Activity	Benefit
Use the online interface to generate JCL and control statements for batch jobs	Simplifies the task of building batch jobs and reduces errors
Initiate DB2 utilities	Simplifies procedures, scheduling, and management tasks associated with running the DB2 utilities or their OEM equivalents
Perform a sequence of actions associated with DB2 utilities	Simplifies complex procedures by generating various job instructions and commands for DB2 utilities in one job
Invoke a user exit	Simplifies customized processing based on conditions encountered during a batch run.

Profiles for Batch Jobs

Overview

This unit presents background information the kinds of profiles !DB/SMU uses when creating batch jobs.

Types of profiles used with batch jobs

Batch jobs use four types of profiles that enable you to tailor the execution of your batch jobs to fill specific needs. Profiles provide the means to define, save, and reuse different instructions and operating conditions for batch jobs.

- Run profiles can be viewed as a type of filtering mechanism. They define what you want to do and what objects you want to act on. That is, they define the set of DB2 tables spaces and index spaces to be processed by the associated SCAN, SPACE, or MONITOR command.
- Exception profiles identify specific conditions that, if encountered during a scan, trigger reporting and the optional execution of specific DB2 utilities or other actions.
- Action profiles create composite or macro utilities to perform a sequence of actions during a batch scan. (A composite or macro utility is one that generates JCL to execute multiple DB2 utilities.) These actions can include a combination of DB2 utility operations and job steps. The sequence of actions is triggered if the job encounters an exception condition that has been defined in an exception profile, EXCPTCTL DD statement, or exception user exit.
- Utility profiles define the operating environment and the options and values required to run the individual DB2 utilities.

Summary of profile functions

This chart describes the functions performed by the profiles that !DB/SMU uses in batch jobs.

Type of Profile	Function
Run profiles	Can be one of three types: *Scan run profile* Identifies the specific spaces to be used for a scan. Generates a batch job stream to scan table spaces and index spaces for performance data. Produces a series of statistics reports. Can also perform SQL updates using the statistics collected for the spaces. *Space map run profile* Identifies the specific spaces to be used for the space map analysis. Generates a batch job stream to analyze space map pages of table spaces and index spaces. Produces a summary report. *Monitor run profile* Monitors the results of a scan or space map analysis for table spaces and index spaces. Identifies those spaces meeting criteria related to dataset size, copy factor, reorg factor, or number of extents. Generates a batch job for the montitor function. Specifies DB2 utility actions and produces selected reports.
Exception profile	Checks data integrity and specifies exception conditions to initiate DB2 utilities, call an exceptions exit, or perform another action. Produces an errors and exceptions report. Executed as an optional choice for a scan run profile.

Summary of profile functions (continued)

Type of Profile	Function
Action profile	Specifies the sequence of steps comprising a job and initiates DB2 utilities. Executed as an optional choice for a monitor or scan run profile. Executed against only those spaces that are scanned by the associated scan run or monitor run profile and meet the exception conditions defined in the associated exception profile.
Utility profile	Provides criteria for running the DB2 utilities The active utility profile determines the criteria for the batch job.

Flow of a Batch Job

Overview

This unit describes the flow of batch jobs for a scan run, a space map run, and a monitor run. It is prerequisite to understanding the way in which the !DB/SMU profiles control job flow.

Finding more information

The flows in this unit are intended as an overview only. Full details about each type of profile are found in the chapters or units referenced in this chart.

Profile Type	Source of Information
Run Profile	"Defining and Using Run Profiles for Batch Jobs" on page 491
Exception Profile	"Introduction to Handling Exception Conditions in Batch Jobs" on page 509
Action Profile	"Specifying Actions Using Action Profiles and Inline User Action Definitions" on page 557
Utility Profile	"General Options for Creating DB2 Utility Profiles" on page 393

Job flow for a space map run

This simple chart illustrates the order in which !DB/SMU uses profiles during a batch space map execution:

Step	Action	Comments
1	Initiate batch space map run using a space map run profile.	The space map run profile defines table spaces or index spaces (or both) against which the space map will run.
2	When running a space map, no additional optional processing can be specified.	To make use of information from a space map run, you will need to execute a monitor run against the results of the space map run.

Job flow for a scan run

This simple chart illustrates the order in which !DB/SMU uses profiles during the batch execution of a scan:

Step	Action	Comments
1	Initiate a batch scan using scan a run profile.	The scan run profile defines the table spaces or index spaces (or both) to be scanned).
2	Optional. Process an exception profile.	This profile defines the conditions under which a particular action will be taken. For example, an action would be taken for those table spaces having a reorg factor greater than 35.
3	Optional. If any objects met the conditions specified in the exception profile, submit the process defined by the action profile	The action profile specifies processing steps to be taken against the objects that met the exception conditions. For example, reorganize the spaces identified in step 2; print the names of the candidate spaces that are to be reorganized and their reorg factor prior to the reorganization.
4	When running the action profile, use the appropriate DB2 utility profiles as needed.	Invoke the DB2 REORG utility to perform any reorganization required in step 3. Apply the criteria specified in the utility profile for the REORG utility.

Job flow for a monitor run

This simple chart illustrates the order in which !DB/SMU uses profiles during a batch monitor run:

Step	Action	Comments
1	Initiate a batch monitor run using the monitor run profile.	The monitor run profile identifies table spaces or index spaces (or both) to be monitored with respect to dataset size, copy factor, reorg factor, and number of extents. The monitor run identifies these table spaces by inspecting the output of a scan or space map execution.
2	Optional. If any objects met the conditions specified in the monitor run profile, submit the process defined by the action profile.	The action profile specifies processing steps to be taken against the objects which met the specified conditions. For example, reorganize the spaces identified in step 1; print the names of the spaces that are to be reorganized and their reorg factor prior to the reorganization.
3	When running the action profile, use the appropriate DB2 utility profiles as needed.	Invoke the DB2 REORG utility to perform any reorganization required in step 2. Apply the criteria specified in the utility profile for the REORG utility.

Considerations when building a monitor job

Candle recommends that you do not scan both table spaces and index spaces, in the same scan.

Generating Batch Jobs

Overview

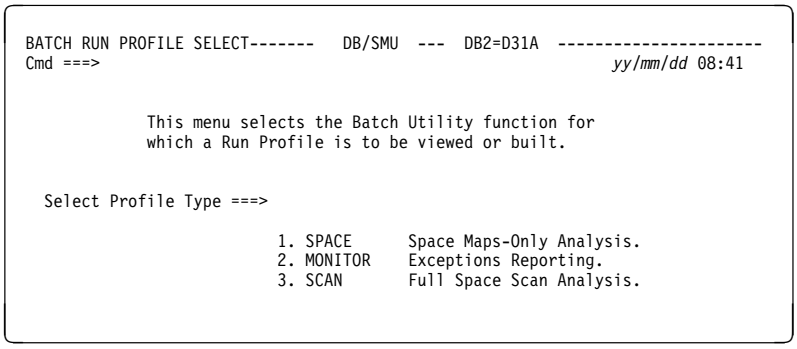
This unit covers the basic procedures for generating batch jobs.

Definition of a batch job

A batch job is the output produced by running the !DB/SMU batch utility. The batch utility consists of the JCL you create using various profiles. Start with a run profile to specify the type of batch job you want to perform (for example, a scan run, space map run, or monitor run) and the spaces (index spaces or table spaces) to be operated on. The run profile can be used alone or in combination with appropriate action or exception profiles. For detailed information on creating run profiles, see "Defining and Using Run Profiles for Batch Jobs" on page 491.

Beginning to build a batch job

Use the following procedures to begin building a batch job.

Step	Action
1	On the Primary Menu, type **5.1** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the Batch Run Profile Select panel. 
3	Select an option for a run profile by typing the number of the run type you want in the Select Profile Type field.
4	Press Enter. **Result**: !DB/SMU displays a list of the existing space map, monitor, or scan profiles on the appropriately named Profiles (for sxample, Scan Profiles) panel.

Generating the batch job stream

Use the following procedures to generate the batch job stream.

Step	Action
1	On the Profile panel, review the list of available profiles. Select the profile you want to use by typing **U** in the SEL field next to its name.
2	Press Enter. **Result**: !DB/SMU displays the Generate Batch Job panel. The entry fileds on this panel vary according to the type of batch job you are building—scan, space, or monitor. A description of the fileds on this panel and how to use can be found in "Elements of the Generate Batch Job Panel" on page 472 GENERATE BATCH JOB ------------- DB/SMU --- DB2=D31A ------------- -------- Cmd ===> *yy/mm/dd* 08:09 Set your values, then press ENTER to generate the jobstream, or END to return. ISPF Skeleton Specification Section: Jobcard Member ===> UTILJOBS (These members are tailored from the ISPF skeleton library defined Utility JCL Member ===> UTILITYU during product installation) Profile Specification Section: Utility Profile ===> Optional. Enter ? for a list. Action Profile ===> Optional. Enter ? for a list. Exception Profile ===> Optional. Enter ? for a list. Update RUNSTATS? ===> N (Y/N)
3	Complete the entry fields on the Generate Batch Job panel.

Generating the batch job stream (continued)

Step	Action
4	Press Enter. **Result**: !DB/SMU processes the entries on the panel, generates the job statements, and displays the Process Member panel.

```
PROCESS MEMBER -----------------   DB/SMU  ---  DB2=D31A  ----------------------
Cmd ===>                                                      yy/mm/dd 08:41


    Batch Utility jobstream has been saved in member TDTDA2U.

    Specify the next action to be performed.

    Select Action ===>

                       1. UPDATE NOW     Submit Batch Utility.
                       2. EDIT           Edit Batch Utility JCL.
                       3. DELETE         Delete the JCL member.
                       4. PRINT          Print the JCL member.
                       5. LOG            Copy JCL member to the Log PDS.

                       6. CONTINUE       Return.
```

Elements of the Generate Batch Job Panel

Overview

This unit describes the elements of the Generate Batch Job panel. You should understand the use of these elements before completing the required fields on the panel.

Use of the ISPF skeleton fields on the Generate Batch Job panel

!DB/SMU uses variables within skeleton members to generate job statements automatically. All run and utility profiles use these skeletons.

Use of the profile specifications on the Generate Batch Job Panel

You can specify several options to use with a monitor or scan run profile, or you can leave these fields blank. Refer to the following chart for an explanation of these options. You can use an option for any monitor or scan profile unless otherwise noted.

Field	Use
Utility Profile	To change the active utility profile, type the name of the profile in this field or type a **?** to display a list for making a selection.
Action Profile	To apply an action profile to this job, type the name of the profile in this field or type a **?** to display a list for making a selection.
Exception Profile (Only applies to scan profiles)	To apply an exception profile to this job, type the name of the profile in this field or type a **?** to display a list for making a selection.
Update RUNSTATS (Only applies to scan profiles)	To perform SQL updates with the statistics, overtype the default N value with **Y**.

Executing the Batch Job You Generated

Overview

Once you have completed the specification of the options !DB/SMU needs to generate the batch JCL, you can choose to take one of several actions against the JCL. This unit describes those actions.

Actions you can take against the generated JCL

Once you have generated the batch JCL, you can take one of these actions.

Step	Action
1	On the Process Member panel, select an action to complete the job. Type one of the following values in the Select Action field: • **1** to submit the batch job • **2** to edit the job statements under ISPF • **3** to delete the job member • **4** to print the job statements • **5** to copy the job member to the log PDS • **6** to display the previous panel and save the job member in the user PDS
2	Press Enter. **Result**: !DB/SMU performs the appropriate action.

Requirements for submitting a !DB/SMU batch job

When submitting a batch job from an online !DB/SMU session, always use the following procedures to submit the job. Submitting a batch job in another manner can result in errors. Do not use the ISPF SUBMIT or SUB command. Doing so may create conflicts in the use of the *userid*||U member in the user PDS.

Step	Action
1	On the Process Member panel, select the EDIT option. Type **2** in the Select Action field.
2	Press Enter. **Result**: !DB/SMU opens the member containing the batch job in ISPF edit mode.

Requirements for submitting a !DB/SMU batch job (continued)

Step	Action
3	Review the job stream for accuracy; make any appropriate changes. When you have completed editing the member, type **END** on the command line or press the appropriate function key. **Result**: The Process Member panel displays.
4	On the Process Member panel, select the Update Now option by typing **1** in the Select Action field.
5	Press Enter. **Result**: !DB/SMU submits the batch job.

Setting or Changing the Active Utility Profile for a Batch Job

Overview

A utility profile provides criteria for running the DB2 utilities. This unit discusses some considerations for using utility profiles and explains the different ways you can designate the active utility profile.

Background about utility profiles

You can define numerous utility profiles, but only one can be *active* at a time. !DB/SMU includes the name of the active utility profile in the job stream it is building. That profile applies to all batch jobs in the job stream until you specify another profile to be the active one.

Reminder about defining utility profiles

See “General Options for Creating DB2 Utility Profiles” on page 393 for information about defining a utility profile.

Setting the active utility profile by updating a run profile

Follow these steps to set the active utility profile by updating a run profile.

Step	Action
1	In the Option field of the !DB/SMU Primary Menu, type **5.1**.
2	Press Enter. **Result**: !DB/SMU displays the Batch Run Profile Select panel.
3	Select the option for monitor or scan run profiles by typing one of the following in the Select Profile Type field: • **2** to select a monitor profile • **3** to select a scan profile
4	Press Enter. **Result**: !DB/SMU displays a list of the existing monitor or scan run profiles.
5	Select the profile you want to use by typing **U** in the SEL field next to its name.
6	Press Enter. **Result**: !DB/SMU displays the Generate Batch Job panel.
7	In the Utility Profile field, identify the utility profile you want to activate using one of the following methods: • Type the name of the profile in the Utility Profile field and complete the remaining fields on the panel. Press Enter. **Result**: !DB/SMU processes the entries on the panel and displays the Process Member panel. The utility profile you specified is included in the JCL that !DB/SMU builds. • Type **?** in the Utility Profile field and complete the remaining fields on the panel. Press Enter. **Result**: !DB/SMU displays a list of profiles. Type **S** in the SEL field next to the name of the profile you want to activate. Press Enter. !DB/SMU processes the panel and displays the Process Member panel. The utility profile you specified is included in the JCL that !DB/SMU builds.

Changing the active utility profile by editing a job stream

You can use the edit option on the Process Member panel to change your created batch utility JCL at any time before you execute it. If you have already generated a job stream for a run profile and want to change the designated utility profile, follow these steps:

Step	Action
1	On the Process Member panel, type **2** in the Select Action field to select the Edit option.
2	Press Enter. **Result**: !DB/SMU opens the member containing the batch job.
3	Scroll to the //SYSIN DD* statement. Insert the following statement before the SCAN, SPACE, or MONITOR statement in the job stream: `UTIL OPTS <profile_name>` where *profile_name* is the name of an exisiting utility profile.
4	Type **END** on the command line or press the appropriate function key. **Result**: !DB/SMU returns to the Process Member panel. The utility profile you designated is now the active utility profile for the job.

Job Statements for the !DB/SMU Batch Utility

Overview

This unit discusses the job statements for the !DB/SMU batch utility.

Automatic generation of the JCL to run the !DB/SMU batch utility

!DB/SMU automatically generates the JCL required to run the !DB/SMU batch utility when you initiate any kind of batch job using a run profile.

Job card

!DB/SMU automatically generates the job card required to run the batch utility.

EXEC statement

!DB/SMU automatically generates the following EXEC statement for a batch job:

```
//UTILT  EXEC  PGM=IKJEFT01,REGION=4M,DYNAMNBR=100
```

DD statements generated by !DB/SMU

This chart describes the DD statements that !DB/SMU generates for batch jobs. *All statements are required unless otherwise indicated.* Those DD statements beginning with the letters **ISP** are required by ISPF and are defined in the IBM ISPF documentation.

DD Statement	Description
ERRORS	Specifies an output file that contains output for all errors and exception conditions. The RECFM=FBA and LRECL=121. If you remove this statement, the report information is directed to the KTSPRINT output file.
EXCPTCTL	Specifies an input-only file for the exception condition statements
ISPCTL0	Used with ISPF Edit SUBMIT, PDS COMPRESS, PDF batch, and for processing file tailoring
ISPFILE	Holds file tailoring output
ISPLLIB	Refers to the !DB/Tools load library

DD statements (continued)

DD Statement	Description
ISPLOG	Refers to the ISPF log
ISPLST1	Used for generating ISPF listings
ISPLST2	Used for generating ISPF listings
ISPMLIB	Refers to the MSGS data set, that contains the error messages issued for the online panels and the batch utility. The ISPMLIB DD statement is optional.
ISPPLIB	Refers to the panels in the !DB/Tools panels library that support batch operations under ISPF
ISPPROF	Refers to the ISPF profile data set
ISPSLIB	Refers to the !DB/Tools skeletons library
ISPTLIB	Refers to the !DB/Tools control library
ISPWRK1	Refers to an ISPF work file
KTSLGPDS	Refers to the !DB/SMU log PDS
KTSPRINT	Receives all of the input created for a batch job. If you want to review the created input, you can print out the contents of KTSPRINT.
KTSSQLER	Refers to the !DB/SMU data set to which messages are written that occur during the following types of processing: • Exception processing for the Days Since Image Copy exception condition • SQL errors that occur during the RUNSTATS catalog update process
MNTFILE	Specifies a sequential file to support a monitor run profile. This file contains 1 record for each DB2 table space and index space that meets the criteria specified by the profile. The LRECL=128 and any multiple of 138 is a valid block size. The //MNTFILE DD statement is not always generated. If you want to create the report it refers to, you must add this statement to the JCL !DB/SMU creates to run its batch utility. You can find more information about the optional sequential file created by the //MNTFILE DD statement in the section “Customized User-Written Reports” on page 631.

DD statements (continued)

DD Statement	Description
REPORTS	Contains the output reports !DB/SMU produces
SFXMLIB	Refers to the !DB/Tools message library
SFXZPARM	Refers to the load library that contains the initialization parameter module (DSNZPARM) for DB2. This module provides the VSAM catalog name for the DB2 catalog and directory spaces.
STATPDS	Refers to the !DB/SMU system PDS
STATREPT	Specifies an output file for the scan batch reports. The RECFM=FBA and LRECL=121. If you remove this statement, the scan statistics are directed to the KTSPRINT output file.
STATUPDS	Refers to the !DB/SMU user PDS
STEPLIB	Refers to the load library containing the !DB/SMU modules
SYSERROR	
SYSIN	Identifies a data set that contains input commands for a batch job. The RECFM=FB and LRECL=80.
SYSPRINT	If you remove the ERRORS DD statement, the SYSPRINT output file receives any error and exception messages. If you remove the STATREPT DD statement, the SYSPRINT output file receives the statistics reports for a scan.
SYSPROC	Names the CLIST library that contains the !DB/SMU CLISTs.
SYSTSIN	Identifies input to the MVS TSO processor
SYSTSPRT	Specifies an output file for TSO transactions and messages
SYSUDUMP	Destination for the MVS ABEND information
XTNTMAP	Prints a list of DASD extents for each DB2 data set that is opened for any kind of processing.

Job card statements

!DB/SMU automatically generates the job card statements. It uses the values from the active utility profile. See "General Options for a Utility Profile" on page 400 for information on defining job card statements in a utility profile. If no utility profile is designated for a job, !DB/SMU supplies default job card statements from the UTILJOBC member in the SKELS library.

Commands for Batch Jobs

Overview

This unit reviews the batch commands that you can use with !DB/SMU. It also includes rules that apply when you manually enter a batch command into a generated job stream.

Background on using commands in batch jobs

!DB/SMU automatically generates the JCL used to run the !DB/SMU batch utility. Contained in that JCL is the batch command to run the function you selected —scan, space map, or monitor. If you want to include other batch commands, you must enter these into the JCL manually.

Batch commands for the space map, monitor, and scan functions

One of these batch commands is automatically included in the !DB/SMU batch utility JCL.

- SCAN—Perform a scan
- SCANR—Perform a scan and generate the SQL statements to update the DB2 catalog with the collected statistics (RUNSTATS)
- SPACE—Perform a space map analysis
- MONITOR—Perform a monitor run

The execution of the batch utility JCL includes the execution of the command to initiate the appropriate run profile.

Batch commands you can enter manually

You can manually add the following commands to generated batch jobs:

- DOCOMPLETEJOBS—Bypass checking for the generation of “ambiguous” utilities
- DUMP—Produce a hex dump of a table space or index page
- FORMAT—Format a range of DB2 table space or index space pages and prints this output describing the page structure
- STACK—Defer the initiation of a DB2 utility until the stack limit is reached

Information on the syntax of these commands and examples of their use are found in “Batch Commands You Can Enter Manually” on page 641.

Rules for manually entering batch commands

Observe these rules when you manually add commands to a generated batch job stream.

- Specify the batch command following the SYSIN statement of a generated job.
- Write the entire command as a single record. (No continuations are allowed; !DB/SMU scans only columns 1–71.)
- Include 1 or more blanks between a command and its operands.
- Enclose keyword operands in parentheses.
- Use blanks to separate keywords from other keywords.
- Add an asterisk (*) in column 1 to indicate a comment.

Building and Executing Batch Jobs in Parallel

Overview

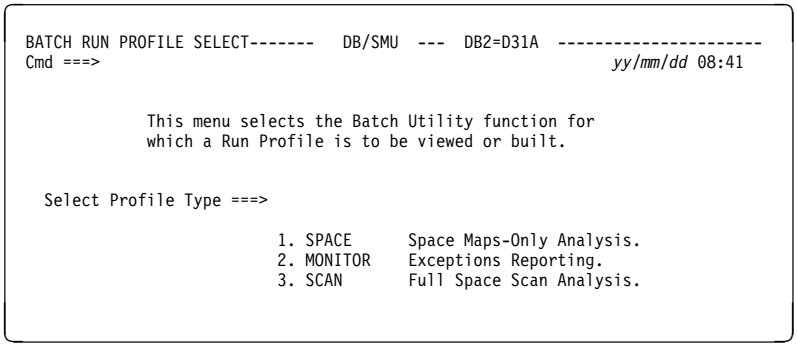
When using !DB/SMU, you can create JCL that permits you to run multiple batch jobs in parallel. This unit addresses the reasons why you might want to create parallel batch jobs, describes the procedure to create the JCL to run the jobs, and provides examples you can use.

Purpose of running parallel batch jobs

Create and execute batch jobs in parallel to reduce the amount of elapsed time required to execute a single long-running job. An example of such a long-running job might be an instance where you want to scan all of the table spaces assigned to a given database. You can break this scan into multiple parallel scans using the procedures in this unit. An example is provided later in this unit to assist you in setting up the parallel jobs.

Beginning to build parallel batch jobs

Use the following procedures to begin building a job that can be executed in parallel.

Step	Action
1	On the Primary Menu, type **5.1** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the Batch Run Profile Select panel. 
3	Select an option for a run profile by typing one of the following in the Select Profile Type field: • **1** for space map run profiles • **2** for monitor run profiles • **3** for scan run profiles

Beginning to build parallel batch jobs (continued)

Step	Action
4	Press Enter. **Result**: !DB/SMU displays the appropriate Profiles panel. On that panel, choose the option to create a new run profile.
5	Press Enter. **Result:** !DB/SMU displays the Run Profile Add panel that permits you to define a new run profile. RUN PROFILE ADD ---------- DB/SMU --- DB2=DB31 ---------- Cmd ===> *yy/mm/dd* 12:38 This screen allows you to specify which table spaces and index spaces are to be included in the SPACE MAP ANALYSIS Run Profile named ===> samprun (Name the new profile) The name fields can be mask values using "*". Dbdname ===> Database name. Space name ===> Table space or index space name. Creator ID ===> Creator Authorization name. VSAM Catalog ===> DB2 dataset high-level qualifier. Data Type ===> (T) Tablespaces (I) Indexspace (B) Both. HSM Auto-Recall ===> N (Y) Yes or (N) No, to issue DFHSM recalls. Scan Thresholds ===> (Y or N) to use intelligent scan LIST ===> N (Y or N) There are two alternatives. LIST N Specify any or all of the fields above. Any table space or index space that matches the above specifications at run time will be selected. Spaces defined later are automatically included. LIST Y Specify any or all of the fields above. Names of eligible spaces will be displayed. You identify now all the spaces you want to be selected This run profile will be for one of the parallel jobs. An example of defining run profiles for parallel jobs is found in “Defining run profiles for parallel jobs” later in this unit.
6	When you have completed the run profile definition, press Enter. **Result:** !DB/SMU re-displays the Profiles panel with the new run profile you defined contained in the list.
7	Select the new run profile by typing **U** in its SEL field.

Beginning to build parallel batch jobs (continued)

Step	Action
8	Press Enter. **Result**: !DB/SMU displays the Generate Batch Job panel. The entry fields on this panel vary according to the type of batch job you are building—scan, space, or monitor. A description of the fields on this panel and how to use them can be found in "Elements of the Generate Batch Job Panel" on page 472.
9	To generate parallel batch jobs, you *must* specify **UTILITYS** in the Utility JCL Member field of the ISPF Skeleton Specification section of the Generate Batch Job panel. The reason for this choice is discussed in "Using the ISPF Skeleton Specification field value UTILITYS" later in this unit. GENERATE BATCH JOB ------------- DB/SMU --- DB2=D31A ------------- -------- Cmd ===> *yy/mm/dd* 08:09 Set your values, then press ENTER to generate the jobstream, or END to return. ISPF Skeleton Specification Section: Jobcard Member ===> UTILJOBS (These members are tailored from the ISPF skeleton library defined Utility JCL Member ===> UTILITYS during product installation) Profile Specification Section: Utility Profile ===> Optional. Enter ? for a list. Action Profile ===> Optional. Enter ? for a list. Exception Profile ===> Optional. Enter ? for a list. Update RUNSTATS? ===> N (Y/N)
10	Press Enter. **Result**: !DB/SMU processes the entries on the panel, generates the job statements, and displays the Process Member panel. PROCESS MEMBER ----------------- DB/SMU --- DB2=D31A ---------- Cmd ===> *yy/mm/dd* 08:41 Batch Utility jobstream has been saved in member TDTDA2U. Specify the next action to be performed. Select Action ===> 1. UPDATE NOW Submit Batch Utility. 2. EDIT Edit Batch Utility JCL. 3. DELETE Delete the JCL member. 4. PRINT Print the JCL member. 5. LOG Copy JCL member to the Log PDS. 6. CONTINUE Return.

Using the ISPF Skeleton Specification field value UTILITYS

When generating the JCL to run a batch job, UTILITYU is the default value in the ISPF Skeleton Specification field of the Generate Batch Job panel. !DB/SMU uses the skeleton identified by UTILITYU to generate the JCL required to run its batch utililty. This skeleton is defined to use member USERID||U in the user PDS as a work area.

When running parallel batch jobs, the use of USERID||U can cause contention. To resolve this, enter the value UTILITYS in the ISPF Skeleton Specification field instead of UTILITYU. This results in !DB/SMU using an alternate skeleton. The alternate skeleton is defined to use a system temporary dataset for a work area in place of the user PDS. Each parallel batch job uses a unique system temporary dataset and thus eliminates contention.

Actions to take against the generated JCL to run the batch utility

Once you have generated the JCL to run the !DB/SMU batch utility, you can take any of the actions shown on the Process Member panel—for example submitting the JCL for execution. Most commonly, when you are creating JCL for parallel executions, you will choose to save the JCL you created for submission by a job scheduler. These steps reflect this more common scenario.

Step	Action
1	On the Process Member panel, select the EDIT option. Type **2** in the Select Action field.
2	Press Enter. **Result**: !DB/SMU displays the JCL to run the !DB/SMU batch utility.
3	Perform these steps against the displayed JCL: 1. Review the job stream for accuracy; make any appropriate changes. 2. Using ISPF commands, save the member under a name of your choice in a partitioned dataset. 3. Type **END** on the command line or press the appropriate function key to return to the Process Member panel. 4. Exit !DB/SMU.

Copying the generated JCL to additional PDS members

Once you have saved the batch JCL in a PDS member, you can clone that member as many times as necessary. You will need as many clones as the number of jobs you want to run in parallel. Once you have copied the JCL to additional members, you must make the following changes to each cloned member:

- Ensure that each clone uses a different run profile. More information about this item can be found in the example provided below.
- Ensure that the JOB statement for each clone uses a unique job name. Unless you do this, the generated JCL will "single-thread" (that is, run sequentially).

Defining run profiles for parallel jobs

This example assists in understanding the need to define multiple run profiles when you want to run parallel batch jobs. You must create a unique run profile for each job that will run in parallel.

Assume that, for a given set of tablespaces whose names begin with the letters SAMPL (for example SAMPLE1, SAMPLE2, SAMPLE3...), you want to scan the databases to which they are assigned. You could perform this job as a single batch scan by filling out the scan run profile definition as shown below.

```
RUN PROFILE ADD ----------------   DB/SMU  ---  DB2=DB31  ----------------------
Cmd ===>                                                    YY/MM/DD  12:38

    This screen allows you to specify which table spaces and index spaces
    are to be included in the SPACE MAP ANALYSIS Run Profile
    named ===> samprun  (Name the new profile)
                    The name fields can be mask values using "*".
         Dbdname ===> *           Database name.
       Space name ===> sampl*     Table space or index space name.
       Creator ID ===>            Creator Authorization name.
    VSAM Catalog ===>             DB2 dataset high-level qualifier.
       Data Type ===> T           (T) Tablespaces   (I) Indexspace   (B) Both.
 HSM Auto-Recall ===> N           (Y) Yes or (N) No, to issue DFHSM recalls.
      .
      .
      .
```

Defining run profiles for parallel jobs (continued)

Now, assume you want to split this job into multiple jobs. You still want to use the same set of table spaces. However, now you will create separate run profiles to separate the scans for those databases whose names start with A, those whose names start with B, and so forth. An example is shown next.

```
RUN PROFILE ADD ----------------   DB/SMU  ---  DB2=DB31  ----------------------
Cmd ===>                                                      yy/mm/dd  12:38

    This screen allows you to specify which table spaces and index spaces
    are to be included in the SPACE MAP ANALYSIS Run Profile
    named ===> sampruna (Name the new profile)
                     The name fields can be mask values using "*".
         Dbdname ===> a*          Data Base name.
       Space name ===> sampl*     Table space or index space name.
       Creator ID ===>            Creator Authorization name.
    VSAM Catalog ===>             DB2 dataset high-level qualifier.
       Data Type ===> T           (T) Tablespaces   (I) Indexspace   (B) Both.
 HSM Auto-Recall ===> N           (Y) Yes or (N) No, to issue DFHSM recalls.
      .
      .
      .
```

Additional run profiles, each having a unique name and each applying to databases whose names start with a different letter, can be created as in this example:

Dbdname	Space Name
a*	sampl*
b*	sampl*
c*	sampl*
d*	sampl*
e*	sampl*
f*	sampl*
.	.
.	.
.	.

Elapsed time when running multiple parallel jobs

The elapsed time to run multiple parallel jobs rather than a single long-running job typically takes the time approximately equal to the elapsed time of the longest-running parallel job.

Saving the JCL created by the !DB/SMU batch utility

In order to save the JCL that !DB/SMU creates to run the DB2 utilities, you must include a step in your action profile that performs a PDSCOPY to a member in a permanent dataset. An alternate way to save the JCL is to perform an IEBCOPY to a permanent dataset.

Chapter 25. Defining and Using Run Profiles for Batch Jobs

Introduction

A run profile identifies the DB2 table spaces and index spaces to be processed by the !DB/SMU batch utility during a monitor, scan, or space map run. This chapter explains how to define and use run profiles for batch jobs.

Before reading this chapter, be sure that you understand the concepts presented in the chapter "Introduction to Batch Operations for !DB/SMU" on page 455.

Chapter Contents

Introduction to Using a Run Profile

Overview

This unit provides general information about using a run profile.

Background about using a run profile

All batch jobs originate with a run profile. A run profile identifies:

- the type of operation !DB/SMU is to perform
- the spaces that are to be processed during a given run of the !DB/SMU batch utility

  The criteria you specify when defining the run profile identify the set of data spaces to be processed. The set can include any combination of table spaces and index spaces.

Uses of run profiles

Use a run profile to perform the following functions for a batch job:

- generate job statements and commands for the batch scan, space map, or monitor functions
- automate procedures for running the batch utility
- initiate DB2 utilities
- produce a variety of reports

Types of Batch Run Profiles

Overview

This unit describes the types of run profiles you can create and the options you can include in the run profile.

Types of run profiles

!DB/SMU uses three types of run profiles. The chart below summarizes the options you can include when using a specific type of run profile to execute the !DB/SMU batch utility.

Type of Run Profile	Options for Generating a Job
Scan	You can • set the active utility profile • include an optional action profile • include an optional exception profile • perform optional SQL updates with the collected RUNSTATS statistics
Space map	None
Monitor	You can • set the active utility profile • include an optional action profile • initiate the following optional DB2 utilities: – COPY (incremental or full) – RECOVERY – REORG – RUNSTATS • print optional batch reports

Background about Creating Run Profiles

Overview

This unit contains background information you need to select candidate spaces and create a run profile.

Background about selecting spaces

The procedure to select spaces is common to both table spaces and index spaces. You can use *specific* names or *generic* names to identify the spaces to be used by a run profile.

- A specific name is the exact name identifying a single space; for example, TESTSPC, or TABL2.
- A generic name is one in which a masking character replaces one or more characters of the name. In this case, the selected spaces are all those whose names match the specified mask. For example, TABL* identifies all the spaces whose name begins with the characters TABL.

Background about using pattern masks

When you provide the name of a table space or index space, you can use special characters that mask some or all of the characters in the name. This feature lets you use generic names that apply to a wide range of existing table spaces and index spaces. Masking also permits you to automatically apply the profile to any new spaces you may add in the future. See “Using Masks to Create Selection Lists” on page 84 for details.

Background about a monitor run profile

Unlike a scan or a space map analysis, a monitor profile run does not process spaces. Rather, monitor profile reviews reports that !DB/SMU produces as the result of a scan or space map analysis and stores in the system PDS. The monitor process selects from these reports the ones it will act on based on criteria you specify when generating a monitor run profile.

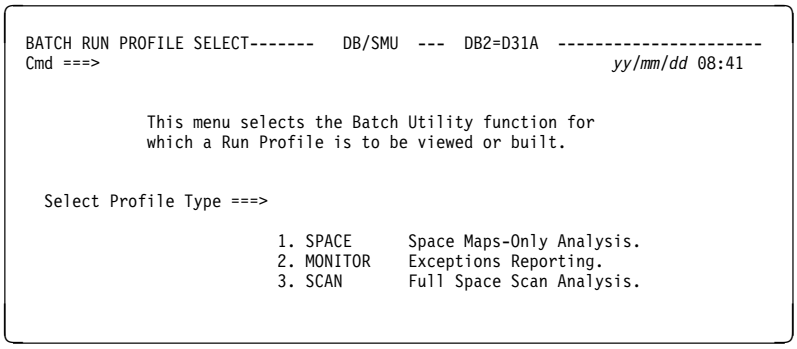
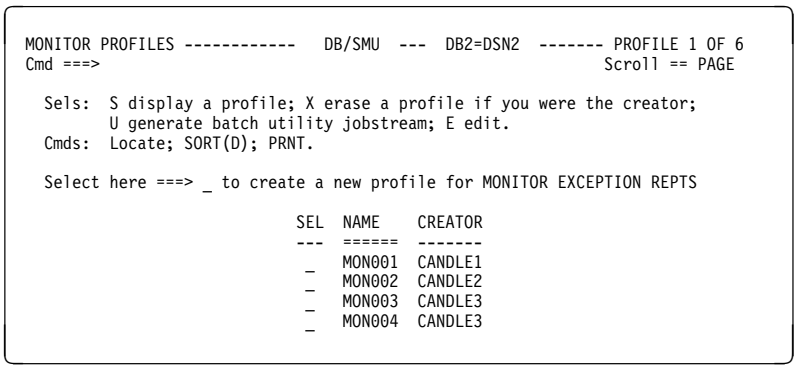
Creating Run Profiles

Overview

This unit provides the procedures for creating a run profile.

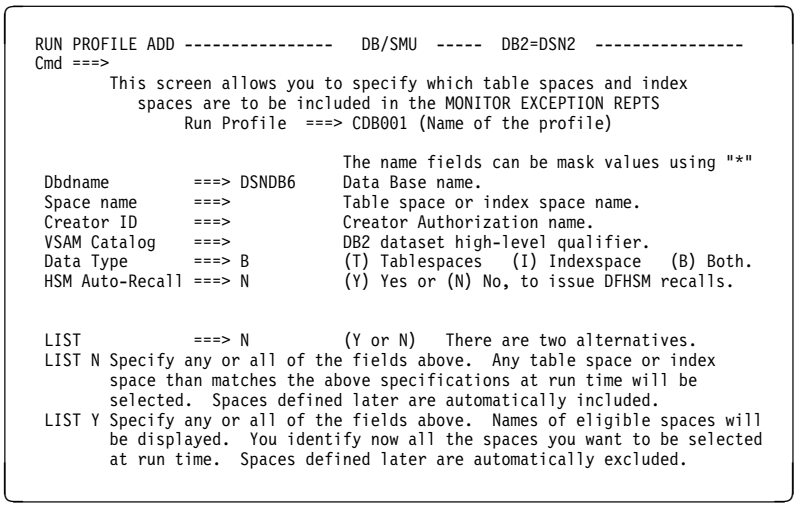
Beginning the creation of a run profile

Use these procedures below to create any type of run profile.

Step	Action
1	On the Primary Menu, type **5.1** in the Option field
2	Press Enter. **Result**: !DB/SMU displays the Batch Run Profile Select panel. 
3	To select an option for a run profile, type its number in the Select Profile Type field:
4	Press Enter. **Result**: !DB/SMU displays a list of the existing space map, monitor, or scan profiles. (This example lists monitor run profiles.) 

Selecting spaces for a run profile

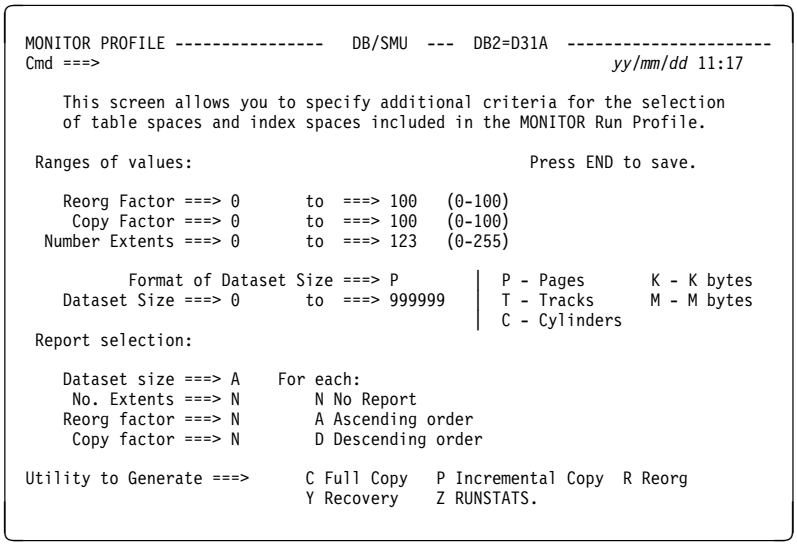
Use the following procedures to select spaces for any type of run profile.

Step	Action
1	On the Space, Monitor, or Scan Profiles panel, type any character in the Select Here field.
2	Press Enter. **Result**: !DB/SMU displays the Run Profile Add panel. 
3	Type a name for your profile in the field provided. The name can be from 1–6 characters.
4	Fill in those entry fields that apply to your profile definition. Type any information required for your profile. Also specify the appropriate List option (**Y** or **N**). The unit “Filling Out the Run Profile Add Panel” on page 499 explains each entry field. This unit also discusses the effect of the List options.
5	If you are creating a scan run profile or space map run profile, your definition is now complete. If you are creating a monitor run profile, continue with the next step of your definition.

```
RUN PROFILE ADD ----------------   DB/SMU  -----  DB2=DSN2  ----------------
Cmd ===>
        This screen allows you to specify which table spaces and index
           spaces are to be included in the MONITOR EXCEPTION REPTS
                Run Profile  ===> CDB001 (Name of the profile)

                                    The name fields can be mask values using "*"
 Dbdname          ===> DSNDB6      Data Base name.
 Space name       ===>             Table space or index space name.
 Creator ID       ===>             Creator Authorization name.
 VSAM Catalog     ===>             DB2 dataset high-level qualifier.
 Data Type        ===> B           (T) Tablespaces   (I) Indexspace   (B) Both.
 HSM Auto-Recall ===> N            (Y) Yes or (N) No, to issue DFHSM recalls.


 LIST             ===> N           (Y or N)   There are two alternatives.
 LIST N Specify any or all of the fields above.  Any table space or index
        space than matches the above specifications at run time will be
        selected.  Spaces defined later are automatically included.
 LIST Y Specify any or all of the fields above.  Names of eligible spaces will
        be displayed.  You identify now all the spaces you want to be selected
        at run time.  Spaces defined later are automatically excluded.
```

Completing the creation of a monitor run profile

Unlike the scan and space map processes for creating a run profile, the process to create a monitor run profile requires the following additional steps for completion.

Step	Action
1	Complete the common procedures for creating a run profile. • If you specified LIST N, !DB/SMU displays the Space, Monitor, or Scan Profile panel. A sample Monitor Profile panel appears below. • If you specified LIST Y, !DB/SMU displays the Table Space Selection List panel. Select the spaces you want to use for this profile according to the instructions in "Selecting candidate spaces from a table space selection list" on page 500. Then type END on the Command line or press the appropriate function key. **Result**: !DB/SMU displays the Space, Monitor, or Scan Profile panel.  MONITOR PROFILE ---------- DB/SMU --- DB2=D31A ---------- Cmd ===> *yy/mm/dd* 11:17 This screen allows you to specify additional criteria for the selection of table spaces and index spaces included in the MONITOR Run Profile. Ranges of values: Press END to save. Reorg Factor ===> 0 to ===> 100 (0-100) Copy Factor ===> 0 to ===> 100 (0-100) Number Extents ===> 0 to ===> 123 (0-255) Format of Dataset Size ===> P | P - Pages K - K bytes Dataset Size ===> 0 to ===> 999999 | T - Tracks M - M bytes | C - Cylinders Report selection: Dataset size ===> A For each: No. Extents ===> N N No Report Reorg factor ===> N A Ascending order Copy factor ===> N D Descending order Utility to Generate ===> C Full Copy P Incremental Copy R Reorg Y Recovery Z RUNSTATS.
2	Optional. Specify values for the reorg factor, copy factor, and number of extents.
3	Optional. Specify values for the dataset format and size. If you do not want to limit the dataset size, set both the lower and upper limits to 0.

Completing the creation of a monitor run profile (continued)

Step	Action
4	Optional. Specify a value for each report type (dataset size, number of extents, reorg factor, and copy factor). For each report, enter one of the following values in the appropriate field: • Type **N** to omit printing of a given report. • Type **A** to print a given report in ascending order. • Type **D** to print a given report in descending order. If you specify **N** for all four report types, !DB/SMU prints a default report, the dataset size report, in ascending order.
5	Optional. Select a utility to run. Type one of the following values in the Utility to Generate field: • **C** to run a full image copy • **P** to run an incremental image copy • **R** to run the REORG utility • **Y** to run the RECOVERY utility • **Z** to run the RUNSTATS utility
6	Save the monitor run profile by typing **END** on the command line or press the appropriate function key. **Result**: !DB/SMU saves the profile you have defined and re-displays the Space, Monitor, or Scan Run Profiles panel.

Filling Out the Run Profile Add Panel

Overview

This unit describes the fields on the Run Profile Add panel and assists you in completing them.

Completing the entry fields

This chart below describes completing the entry fields on the Run Profile Add panel. *Entries are required unless indicated otherwise.*

Field	Use
Dbdname	Enter a specific database name to select spaces in only 1 database; enter a generic name to select spaces in a group of databases.
Space Name (optional)	Enter a specific table space or index space name to select 1 space; enter a generic name to select a range of spaces.
Creator ID (optional)	Enter a specific creator ID to select spaces for 1 ID; enter a generic name to select spaces for a range of IDs.
VSAM Catalog (optional)	Enter a specific name to select spaces with a specific high-level qualifier; enter a generic name to select spaces in a range of high-level qualifiers.
Data Type	• Specify **T** to select table spaces. • Specify **I** to select index spaces. • Specify **B** to select both table spaces and index spaces.
HSM Auto-Recall	If your site uses DFHSM, this field determines whether (**Y**) or not (**N**) to recall any migrated datasets that fall within the scope of the profile.
List	Optional. Displays a selection list of spaces. Information about using the List field appears in "Using the List field to select spaces" on page 500.

Using the List field to select spaces

To use the List field, follow the guidelines below.

Value Entered in the List Field	Result
N to include any existing or future table spaces or index spaces that match the naming criteria	!DB/SMU displays the Space, Monitor, or Scan Profiles panel. This panel now displays the name of the new profile you have defined.
Y to display the appropriate Space Selection List panel, a list of specific table spaces or index spaces. This option excludes any future spaces from the profile definition.	!DB/SMU displays a selection list of spaces. Use this list to complete your space selections. (See "Selecting candidate spaces from a table space selection list" for additional instructions.)

Selecting candidate spaces from a table space selection list

If you typed **Y** in the List field on the Run Profile Add panel, you can now select spaces from selection list to which the profile will apply. Review this chart for instructions using the Table Space Selection List panel.

Task You Want to Perform	Procedure
Select a candidate space from the list	Type **S** in the SEL field next to the name of the space.
Include additional databases in the list.	Type a specific or generic name in the DBD field.
Include additional types of spaces in the list	Do one of the following: • Type **T** to include additional table spaces. • Type **I** to include additional index spaces. • Type **B** to include additional table spaces and index spaces.
Include spaces with additional creator IDs in the list.	Type a specific or generic name in the CR8 field.
Include spaces with additional catalog name in the list	Type a specific or generic name in the CAT field.
Include additional table space or index space names in the list	Type a specific or generic name in the SPC field.
Complete the selection process	Type **END** on the command line or press the appropriate function key.

Managing Existing Run Profiles

Overview

This unit discusses the methods for you can use to manage run profiles.

Managing run profiles

Use the Run Profiles panel to manage existing profiles for the batch space map, monitor, and scan functions.

Task You Want to Perform	Procedure to Follow
Display the definitions in an existing run profile	Type **S** in the SEL field next to the name of the profile and press Enter. **Result**: !DB/SMU displays the Profiles panel. The panel title identifies the profile type, for example, Monitor Profiles or Space Profiles.
Erase an existing run profile	Type **X** in the SEL field next to the name of the profile you want to erase and press Enter. **Result**: !DB/SMU erases the name from the list and deletes the member containing the profile from the system PDS.
Generate the job stream required to execute the !DB/SMU batch utility	Type **U** in the SEL field next to the name of the profile and press Enter. **Result**: !DB/SMU generates the required job stream.
Edit an existing run profile	Type **E** in the SEL field next to the name of the profile you want to edit and press Enter. *To edit a dataset, you must be the creator of the dataset.* **Result**: !DB/SMU displays the Run Profile Edit panel.
Generate the job stream required to execute the !DB/SMU batch utility	Type **U** in the SEL field next to the name of the profile and press Enter. **Result**: !DB/SMU generates the required job stream.

Managing run profiles (continued)

Task You Want to Perform	Procedure to Follow
Locate the name of a profile in the profile list	Type **L** *profilename* on the command line and press Enter. **Result**: !DB/SMU locates the named profile in the list. If no profile exists with the name you entered, !DB/SMU positions the cursor in the list at a point just prior to where the named profile might occur.

Name under which a run profile is stored in the system PDS

When !DB/SMU stores a run profile in the system PDS, it adds a 2-character prefix to the name. For example, if you assign the name MON001 to a monitor run profile, !DB/SMU stores it under the name, ZMMON001. The following chart explains the prefix codes.

Prefix Character	Explanation
First character	Z is always the first character.
Second character	Defines the type of profile as follows: **M** for a monitor run profile **S** for a scan run profile **X** for a space map run profile

Viewing Information about a Run Profile

Overview

This unit contains instructions for viewing information about run profiles. The information can also be accessed online.

Background about the online information for run profiles

The online information for the scan, space map, and monitor profiles includes the information you entered to create the profile. Information for each of the profile types also includes the date on which the profile was last executed. In addition, the information for a space map profile includes data gathered during the last job. The later sections in this unit describe the fields provided for each profile type.

Accessing the online run profile information

Follow the steps in this chart to view the online run profile information.

Step	Action
1	On the Primary Menu, select option 5, Profiles.
2	Press Enter. **Result:** !DB/SMU displays the Profile Menu.
3	On the Profile Menu, select option 1, Batch.
4	Press Enter. **Result:** !DB/SMU displays the Batch Run Profile Select Menu.
5	On the Batch Run Profile Select menu, select the type of run profile you want to look at (scan, space map, or monitor).
6	Press Enter. **Result:** !DB/SMU displays the Space, Monitor, or Scan Profiles panel.
7	On the profiles panel, select an existing profile from the list using the S select.
8	Press Enter. **Result:** !DB/SMU displays the online information for the type of run profile you selected.

Contents of an online run profile for a space map

The first part of the run profile display contains the profile definition. The second part contains the data from the most recent use of the profile. The online information differs from the printed batch report. It summarizes all the spaces processed by the job.

Data fields in part 1 of the space map run profile online summary report

The chart explains the data fields in part 1 of an online report for a space map run profile.

Field	Contents
Profile Name	Name of the space map profile
Type	Space Map Analysis identifies the type of run profile
Profile Creator	ID of the profile creator
Dbdname	Value entered in the definition for the database name
Space Name	Value entered in the definition for the space name
Creator ID	Value entered in the definition for the creator ID
VSAM Catalog	Value entered in the definition for the VASM catalog
Data Type	T, I, or B to indicate table, index, or both types of spaces, respectively
HSM Auto-Recall	Value that indicates whether DFHSM recalls were issued during the run
Stats from last run	The name of the job and the date on which it was run

Data fields in part 2 of the space map run profile online summary report

The chart below describes the data fields in part 2 of the space map run profile online summary report.

Field	Contents
Number of Spaces	Total for each type of space
Number of Pages	Total of allocated pages in each type of space
Number of Extents	Total of DASD extents for each dataset
Number of Tracks	Total of DASD tracks for each dataset
Number of Cylinders	Total of DASD cylinders for those datasets that are allocated on cylinder boundaries
Pages Updt/Used	Number of pages updated during the execution
Pct Pages Updt/Used	Percent of pages used during the execution
Free Index Tracks	Number of unused tracks available to hold index data

Data fields in the scan run profile online summary report

The chart explains the data fields shown in the online summary report.

Field	Contents
Profile Name	Name of the scan run profile
Type	DB2 space scan indicator
Profile Creator	ID of the profile creator
Dbdname	Value entered in the definition for the database name
Space Name	Value entered in the definition for the space name
Creator ID	Value entered in the definition for the creator ID
VSAM Catalog	Value entered in the definition for the VASM catalog
Data Type	T, I, or B to indicate table, index, or both types of spaces, respectively
HSM Auto-Recall	Value indicating whether DFHSM recalls were issued during the run
Scan Thresholds	Value indicating whether to use intelligent scan function.

Data fields in the scan run profile online summary report (continued)

Field	Contents
Pages Update Percent	Threshold specified for percent of pages updated when using intelligent scan.
Tracks Update Percent	Threshold specified for percent of tracks updated when using intelligent scan.
Stats from last run	The name of the job and the date on which it was run

Background about the monitor profile online information

The online information for a monitor run profile provides information about the profile definition. It does not provide any information on the latest job. You can display two panels of definition information for a monitor run profile. The first panel contains the generic information. The second panel shows the list of specific spaces to which this profile applies.

Data fields in the monitor run profile online summary report

The chart explains the definition information that is displayed about a monitor run profile. This information is displayed on the first panel of the report.

Field	Contents
Profile Name	Name of the monitor run profile
Type	Monitor exception reports indicator
Profile Creator	ID of the profile creator
Dbdname	Value entered in the definition for the database name
Space Name	Value entered in the definition for the space name
Creator ID	Value entered in the definition for the creator ID
VSAM Catalog	Value provided in the definition for the VASM catalog
Data Type	T, I, or B to indicate table, index, or both types of spaces, respectively
HSM Auto-Recall	Value indicating whether DFHSM recalls were issued during the run
Range Values	Profile specifications for: • copy factor • reorg factor • number of extents • dataset size range • size format for the dataset
Report Chosen	Type of optional reports specified in the profile and whether the request was to print in ascending or descending order.
Stats from last run	The name of the job and the date on which it was run

Data fields in the monitor run profile online summary report (continued)

The second panel of the monitor run profile online summary report is generated when the monitor run profile acts on a list of spaces. This follwowing chart explains the definition information that is displayed about a monitor run profile that is defined for a list of spaces.

Field	Contents
Profile Name	Name of the monitor run profile
Type	Monitor exception reports indicator
Profile Creator	Value given for the profile creator in the definition

Chapter 26. Introduction to Handling Exception Conditions in Batch Jobs

Introduction

Using exception conditions in your batch jobs can help you automate the management of your DB2 system. This chapter provides you with an overview of various types of exception processing—exception profiles, exception control statements, and exception exits. Each type of exception processing is discussed in a separate chapter in this batch section.

Before using this chapter, read "Introduction to Batch Operations for !DB/SMU" on page 455 to gain an understanding of !DB/SMU batch functions.

Chapter Contents

Introduction to Using Exception Conditions

Overview

This unit describes the types of exception conditions you can define and ways in which each of these types of definitions can assist you in monitoring and controlling your DB2 subsystem.

Background about using exception conditions

You can use exception conditions to trigger the execution of various actions to be taken by your !DB/SMU batch job. !DB/SMU provides the ability to define over 70 types of exception conditions. Using these exception conditions in batch jobs permits you to automatically trigger these types of actions:

- Automate DB2 backups, reorganizations, updates to DB2 catalog statistics, and recovery operations by initiating these DB2 utilities:
  - COPY (for full or incremental copies)
  - RECOVER
  - REORG
  - RUNSTATS
- Generate an error and exception report. !DB/SMU produces an error and exception report each time you run a batch scan.
- Combine DB2 utility operations and reporting
- Perform proactive performance tuning
- Invoke a user exit
- Provide criteria to trigger an action profile

Flexibility in defining exception conditions

You have great flexibility in defining exception conditions. You can trigger actions resulting from conditions that are always true or rarely true. Likewise, you can trigger actions for your entire system or for a single space.

Examples of using exception conditions to automate your system

These examples illustrate the use of exception conditions in batch jobs to help you automate system management.

Automated Activity	Example
Perform a full image copy when the number of days since the last full copy reaches or exceeds a desired threshold	Execute a !DB/SMU batch job and specify exceptions that trigger the execution of the DB2 COPY utility when 7 or more days have passed since the last full copy.
Reorganize selected table spaces when the amount of free space falls below a desired threshold	Execute a !DB/SMU batch job and specify exceptions that trigger the execution of the DB2 REORG utility when the free space falls below 15%.
Update statistics in the DB2 catalog when changes to selected spaces reach or exceed a desired threshold	Execute a !DB/SMU batch job and specify exceptions that trigger the execution of the DB2 RUNSTATS utility when 30% or more of the pages have been updated.
Recover selected table and index spaces when the number of error pages reaches or exceeds a desired threshold	Execute a batch run profile and specify exceptions that trigger the execution of the DB2 Recovery utility when data integrity errors exist in 5% or more of the pages.

Methods for Specifying Exception Conditions

Overview

This unit explains the options you can use to specify exception conditions.

How to specify exception conditions

You can use any of the following methods to specify exception conditions.

Method	Explanation	Procedure
Use an exception profile. An exception profile is available *only* for a batch scan.	Exception profiles contain conditions, values, and actions. You use the online exception profile definition facility to specify the conditions and resultant actions.	Specify the name of an exception profile on the Generate Batch Job panel when you generate a job stream for a scan run profile.
Use exception control statements	Exception control statements define exception conditions. You can add exception control statements to any batch job. The exception control statements can be entered into the JCL inline following the EXCPTCTL DD statement, or can be included in a dataset such as a PDS member.	Edit the generated job stream for a scan run profile and add exception control statements following the EXCPTCTL DD statement, or name the dataset containing the exception control statements as a parameter on the EXCPTCTL DD statement.
Use an exception exit	A user-written exception exit extends your ability to define exception conditions and specify spaces against which actions are to be taken. See the next section.	Invoke the exit by specifying conditions for it in an exception profile or by adding exception control statements following the EXCPTCTL DD statement.

Extensive exception processing requires an exception exit

An exception exit provides a level of processing that is more extensive than that which is achievable by using keyword-only criteria such as used in an exception profile or exception control statements.

Thus, if you want to vary the conditions by which a set of actions is taken, or want to have programatic control over the spaces affected by an exception condition, you should invoke a user-written exception exit. For information about circumstances under which you might want to use an exception exit, see "Background on Using Exception Exits" on page 542.

Sources of information

Subsequent chapters describe the methods defined in the preceding chart. Refer to them for additional information.

Method of Defining Exception Conditions	Source of Additional Information
Exception profile	"Using Exception Profiles" on page 517
Exception control statements (either inline in the JCL or contained in an exception control member)	"Using Exception Control Statements and Exception Control Members" on page 529
Exception exit	"Using Exception Exits" on page 541

Order of Processing for Exception Conditions

Overview

To use exception conditions effectively requires an understanding of how !DB/SMU processes these conditions. This unit provides that information.

Relationship of exception conditions to a utility profile

The active utility profile for the batch job specifies how the DB2 utilities for batch jobs are implemented.

Using exception conditions in an exception profile versus in exception control statements

Caution: If exception conditions are present in the batch JCL in both inline exception control statements and in an exception profile, then the conditions in the inline exception control statements override the exception conditions in the exception profile.

Levels of significance for exception actions

The following chart ranks exception actions in their order of significance. The most significant is 1; the least 6.

Level of Significance	Exception Action	Function
1	RECOVER	Recover the space
2	REORG	Reorganize the space
2	REORGIX	If the object is an index, reorganize it. If the object is a table space, reorganize the associated indexes.
2	REORGTS	If the object is a table space, reorganize it. If the object is an index, reorganize the associated table space.
3	FULLCOPY	Perform a full image copy
4	INCRCOPY	Perform a partial image copy
5	RUNSTATS	Update DB2 catalog statistics
6	SKIP	Do not print the error and exception report

How !DB/SMU processes exception conditions

When !DB/SMU finds that an exception condition is true, it performs the action you specified for that condition. The following chart shows the results of processing a given exception condition:

Exception Condition	Result
A *single* exception condition is true	Triggers the action you have specified
Multiple exception conditions are true	Triggers the most *significant* action; !DB/SMU disregards all other actions. (The exception is the case in which a true exception condition is found in both an exception control statement and an exception profile. In this case, the exception condition in the exception control statement overrides the exception condition found in the exception profile. (See "Using exception conditions in an exception profile versus in exception control statements," above, for information on overrides for exception conditions. See "Levels of significance for exception actions" on page 514 for information on the significance of actions.)
A condition that triggers an exception exit is true	Triggers the exit and any actions it prescribes. !DB/SMU disregards *all* other actions in the profile.
A hit occurs on a condition that lacks a specified action	Triggers *no* action; !DB/SMU simply lists the condition in the errors and exceptional conditions report.

Examples of how significance applies to actions

The following examples show how actions supersede each other according to their levels of significance.

IF conditions are true for both...	THEN...
An image copy and a reorganization of a space	The batch job reorganizes the space. !DB/SMU ignores the image copy action.
A full image copy and an incremental image copy	The batch job performs a full image copy instead of a partial image copy.

Chapter 27. Using Exception Profiles

Introduction

Using exception conditions in your batch jobs can help you automate the management of your DB2 system. This chapter helps you to define exception profiles to assist in controlling !DB/SMU batch jobs.

Before using this chapter, read "Introduction to Batch Operations for !DB/SMU" on page 455 to gain an understanding of !DB/SMU batch functions. Also, read "Introduction to Handling Exception Conditions in Batch Jobs" on page 509 for an overview of defining exception conditions.

Chapter Contents

Creating an Exception Profile

Overview

This unit contains procedures for creating an exception profile.

Advantages of using an exception profile

An exception profile defines conditions that can trigger a report of exceptions and errors and the optional execution of a DB2 utility during a batch scan. Using an exception profile provides these advantages:

- A structured interface for easily creating and maintaining exception profiles
- Extensive online help for defining exception conditions
- A convenient means for saving and applying frequently used exception conditions

Specifying an exception profile for a batch job

When you choose a run profile for batch scans, you can also specify an exception profile. See “Generating Batch Jobs” on page 469 for these procedures.

Editing an exception profile

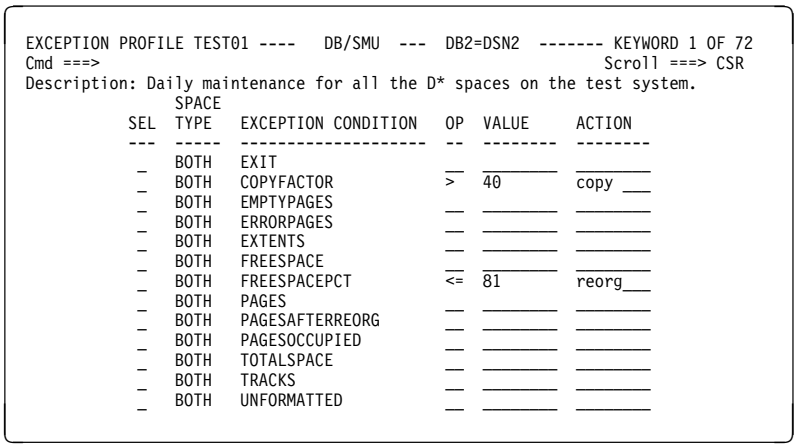
You cannot edit a exception profile using an external editor such as ISPF.

Overriding an exception profile

To modify or override the conditions in an exception profile for a specific batch job, you can add the necessary statements in an exception control member or after the EXCPTCTL DD statement in the generated job stream. You can find information about exception control statements and creating an exception control member in “Using Exception Control Statements and Exception Control Members” on page 529.

Procedures for creating an exception profile

Follow these steps to create an exception profile.

Step	Action
1	On the Primary Menu, type **5.2** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the Exception Profiles panel.
3	Move the cursor to the Select here field and type **S** in the field. **Result**: The cursor advances to the Name field.
4	Type your choice for a profile name in the Name field.
5	Press Enter. **Result**: !DB/SMU displays the Exception Profile *profilename* panel. The *profilename* variable in the panel title is the name you gave to the exception profile you are creating. The Exception Profile *profilename* panel is scrollable and lists all the exception conditions you can define for an exception profile.
6	Define the exception conditions you want to include in the profile. Refer to "Background about Defining Exception Conditions in an Exception Profile" on page 520 for details on creating definitions. 
7	When you are finished, exit the panel by typing one of these commands on the command line: • **SAVE** to save the profile • **QUIT** to exit without saving the profile **Result**: !DB/SMU displays the Exception Profiles panel.

```
EXCEPTION PROFILE TEST01 ----   DB/SMU  ---  DB2=DSN2  ------- KEYWORD 1 OF 72
Cmd ===>                                                    Scroll ===> CSR
Description: Daily maintenance for all the D* spaces on the test system.
                SPACE
           SEL  TYPE   EXCEPTION CONDITION   OP  VALUE     ACTION
           ---  -----  --------------------  --  --------  --------
            _   BOTH   EXIT                  __  ________  ________
            _   BOTH   COPYFACTOR            >   40        copy ___
            _   BOTH   EMPTYPAGES            __  ________  ________
            _   BOTH   ERRORPAGES            __  ________  ________
            _   BOTH   EXTENTS               __  ________  ________
            _   BOTH   FREESPACE             __  ________  ________
            _   BOTH   FREESPACEPCT          <=  81        reorg___
            _   BOTH   PAGES                 __  ________  ________
            _   BOTH   PAGESAFTERREORG       __  ________  ________
            _   BOTH   PAGESOCCUPIED         __  ________  ________
            _   BOTH   TOTALSPACE            __  ________  ________
            _   BOTH   TRACKS                __  ________  ________
            _   BOTH   UNFORMATTED           __  ________  ________
```

Background about Defining Exception Conditions in an Exception Profile

Overview

This unit supplies background you need to use the Exception Profile *profilename* panel to define exception conditions in an exception profile.

Contents of an exception profile definition

The scrollable Exception Profile *profilename* panel lists all of the exception conditions you can define and save in a named exception profile. The completed profile definition uses only those conditions that you modify with operators, values, and actions.

Elements of the Exception Profile panel

This is a partial example of an Exception Profile *profilename* panel.

```
EXCEPTION PROFILE TSTR 1 ---   DB/SMU  ---  DB2=DB31  ------- KEYWORD 1 OF 72
Cmd ===>                                                 Scroll ===> PAGE
Description: <--- Optionally enter a description of the profile here.
           SPACE
     SEL   TYPE   EXCEPTION CONDITION   OP  VALUE     ACTION
           2            3               4    5         6
     ---  -----  --------------------  --  --------  --------
      _    BOTH   EXIT                  __  ________  ________
      _    BOTH   COPYFACTOR            __  ___       ________
      _    BOTH   EMPTYPAGES            __  ________  ________
      _    BOTH   ERRORPAGES            __  ________  ________
      _    BOTH   EXTENTS               __  ________  ________
      .
      .
      .
      _    TABLE  AVGLENPERROW          __  ________  ________
      _    TABLE  AVGPAGESPTRREC        __  ________  ________
      _    TABLE  COMPRESSION           __  ________  ________
      .
      .
      .
      _    INDEX  AVGKEYSPERSUBPAGE     __  ________  ________
      _    INDEX  AVGRECORDSPERKEY      __  ________  ________
      _    INDEX  AVGRECORDSPERPAGE     __  ________  ________
      .
      .
      .
      ************** END OF EXCEPTION ACTIONS ***************
```

1 Name you assigned to the exception profile you are defining.

2 Type of space (index space, table space, or both) to which the exception condition applies.

3 Name of the exception condition

Elements of the Exception Profile panel (continued)

4 Relational operator. For more information about the relational operators you can use in this column, see "Specifying a relational operator for an exception condition" in the unit "Filling In the Exception Profile Panel" on page 523.

5 Threshold used to trigger the exception condition or value to control the type of processing that !DB/SMU is to perform. For more information about the entries you can use in this column, see "Specifying a threshold or value for an exception condition" in the unit "Filling In the Exception Profile Panel" on page 523.

6 Optional action !DB/SMU is to take if the condition you specified using the relational operator and value is true. For more information about the optional actions you can specify in this column, see "Specifying an action for an exception condition in an exception profile" in the unit "Filling In the Exception Profile Panel" on page 523.

Organization of the list of exception conditions

The exception conditions in the scrollable list on the Exception Profile *profilename* panel appear in the following order:

- Conditions applicable to both table spaces and index spaces
- Conditions applicable only to table spaces
- Conditions applicable only to index spaces

The Space Type column identifies the type of DB2 space (Both, Table, or Index) to which the condition can apply.

Defining or deleting exception conditions in a profile

This chart provides instructions for defining or deleting exception conditions on the Exception Profile *profilename* panel. Any conditions that you add, remove, or change are highlighted until you save the profile.

Task	Procedure
Add or change a definition for an exception profile.	For the condition you want to apply, type appropriate values in the Op, Value, and Action fields.
Remove a condition from an exception profile.	For the condition you want to delete, blank out the values in the Op, Value, and Action fields.

Displaying help for specific exception conditions

To display help for an exception condition, follow these steps.

Step	Action
1	To display help for an exception condition, type **?** in the Sel field next to the condition. (To sequentially display help for multiple exception coditions, type **?** in all the applicable Sel fields.)
2	Press Enter. **Result**: !DB/SMU displays help for the exception condition you selected in a pop-up window. If you selected help for multiple exception conditions, you can display the help for each exception condition sequentially by continuing to press Enter.

Filling In the Exception Profile Panel

Overview

This unit provides you with the information you need to fill in the various columns of the Exception Profile *profilename* panel.

Specifying a relational operator for an exception condition

This chart lists the relational operators you can enter in the Op field of the Exception Profile *profilename* panel.

Relational Operator	Meaning
>	Greater than
<	Less than
¬=	Not equal to
=	Equal to
>=	Greater than or equal to
=>	Equal to or greater than
<=	Less than or equal to
=<	Equal to or less than

Specifying a threshold or value for an exception condition

These are the entries you can type in the Value column. The entry you make in the Value column depends on the exception condition to which it applies.

Type of Exception Condition	Value To Enter
EXIT condition	Name of an exception exit. When you specify the name of an exit for the EXIT condition, you must use the = (equals) relational operator.
CLUSTERING condition	Whether an index is a clustering index. **Y** for a clustering index, **N** for a nonclustering index, or **D** for a clustering index by default. When you specify the type of index in the Value column, you must use the = (equals) or ¬= (not equals) relational operator.

Specifying a threshold or value for an exception condition (continued)

Type of Exception Condition	Value To Enter
COMPRESSION condition	Whether compression is in effect. **Y** if compression is in effect, **N** for it is not. When you specify compression in the Value column, you must use the = (equals) or ¬= (not equals) relational operator.
Percentage condition	Integer between 0–100. You can use any relational operator.
All other conditions	Integer between 0–99,999,999.

Specifying an action for an exception condition in an exception profile

This chart lists the valid actions you can specify in the Action column for an exception condition used in an exception profile.

Action	Result
COPY	Performs a full image copy
INCRCOPY	Performs an incremental image copy
RECOVER	Recovers the space
REORGIX	If the object is an index, reorganizes it. If the object is a table space, reorganizes the associated indexes.
REORGTS	If the object is a table space, reorganizes it. If the object is an index, reorganizes the associated table space.
REORG	Reorganizes the space
RUNSTATS	Updates the statistics in the DB2 catalog
SKIP	Cancels the printing of the errors and exceptional conditions report

Reminder about processing exception conditions

To review information about how !DB/SMU processes exception conditions and the order of significance of exception conditions, see “Order of Processing for Exception Conditions” on page 514.

Examples of setting triggers in exception profiles

The chart below gives two examples of setting triggers in exception profiles.

Purpose	Example

Set the following trigger values:
• Perform an incremental image copy if the copy factor is high.
• Reorganize the table space if the number of extents is high.
• Reorganize the table if the CLUSTERRATIO is low.
	`COPYFACTOR > 40 COPY`
	`EXTENTS => 25 REORG`
	`CLUSTERRATIO <= 50 REORG`

Invoke the YOSEMITE exit for *any* of the following trigger values:
• 10% of the pages contain updates.
• Pointer records exist.
• The size of the table space is less than 24 pages.
	`EXIT = YOSEMITE`
	`PCTPAGESUPDATED > 10 COPY`
	`POINTERRECORDS > 0 REORG`
	`PAGES <= 24 SKIP`

A note about reorganizing table spaces based on index conditions

There are circumstances where you may want to reorganize table spaces based on a triggering condition you specify for indexes. An example might be, “reorganize those table spaces for which the associated index has a CLUSTERRATIO<80”. It is not possible to perform this processing using an exception profile. Rather, you must write an exception exit, exiting with return code 28 (or 128), to perform this processing. Information about writing exception exits is found in “Using Exception Exits” on page 541.

Accessing and Managing Exception Profiles

Overview

!DB/SMU provides easy access to existing exception profiles. This unit contains instructions for accessing existing profiles using the Exception Profiles panel.

Elements of the Exception Profiles panel

This is an example Exception Profiles panel.

```
EXCEPTION PROFILES ----------   DB/SMU  ---  DB2=DSN2  ------- PROFILE 1 OF 10
Cmd ===>                                                 Scroll ===> CSR
                                                                      SR
         Sels: S display a profile  E erase a profile.
         Cmds: Locate, PRNT, SORT/SORTD, DO.

         The following Exception Profiles are available.
         Select here ===> _ to create a new profile. Name ===>

      PROFILE      CREATED           MODIFIED
 SEL  NAME      USER      DATE     USER      DATE     DESCRIPTION
        1         2         3        4         5        6
 ---  =======  =======  ========  =======  ========  --------------------------
  _   CANDLE   CANDLE1  99/02/21  CANDLE1  99/04/30  Monitor corporate base
  _   CUSTM1   CANDLE1  99/04/10  CANDLE1  99/04/30  Monitor customer base
  _   CLUSTR   CANDLE1  99/04/08  CANDLE1  99/04/30  Check cluster ratio
  _   DASDRW   CANDLE1  99/04/09  CANDLE1  99/04/09  Check ┐ occupied
  _   DASDWS   CANDLE1  99/02/21  CANDLE1  99/05/20  Check wasted DASD space
  _   EMPLY1   CANDLE1  99/04/09  CANDLE1  99/04/30  Monitor employee base
  _   EXIT01   CANDLE1  99/04/11  CANDLE1  99/04/30  Call CLIST exit 01
  _   EXIT02   CANDLE1  99/02/11  CANDLE1  99/05/16  Call REXX exit 02
  _   MNTDAY   CANDLE1  99/04/09  CANDLE1  99/04/30  Daily maintenance
  _   MNTWKL   CANDLE1  99/02/21  CANDLE1  99/05/20  Weekly maintenance
  _   POINTR   CANDLE1  99/04/09  CANDLE1  99/04/30  Check pointer records
  _   PROD01   CANDLE1  99/04/09  CANDLE1  99/04/09  Production system
  _   TEST01   CANDLE1  99/04/08  CANDLE1  99/04/30  Test system
```

1 Name of each existing exception profile

2 User ID of the person who created the profile

3 Date on which the profile was created

4 User ID of the person who last modified the profile

5 Date on which the profile was last modified

6 Optional descriptive text you provide when you create a profile

Sorting the list of exception profiles

You can use the SORT and SORTD commands respectively to sort the list of existing] profiles in ascending or descending order. You can sort on any column except Description.

Accessing existing exception profiles

Follow these steps to access a list of existing exception profiles.

Step	Action
1	On the Primary Menu, type **5.2** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the Exception Profiles panel. This scrollable panel contains a list of all existing exception profiles.

Managing existing exception profiles

This chart provides details of how to use the Exception Profiles panel to manage the existing exception profiles.

Task	Procedure
Display the definitions in an existing exception profile	Type **S** in the SEL field before the name of the profile and press Enter. **Result**: !DB/SMU displays the Exception Profile *profilename* panel.
Erase an existing exception profile	Type **E** in the SEL field before the name of the profile and press Enter. **Result**: !DB/SMU erases the name from the list and deletes the member from the user PDS.

Chapter 28. Using Exception Control Statements and Exception Control Members

Introduction

Using exception conditions in your batch jobs can help you automate the management of your DB2 system. This chapter helps you to define exception control statements to assist in controlling !DB/SMU batch jobs. It also helps you construct an exception control member in a PDS.

Before using this chapter, read "Introduction to Batch Operations for !DB/SMU" on page 455 to gain an understanding of !DB/SMU batch functions. Also, read "Introduction to Handling Exception Conditions in Batch Jobs" on page 509 for an overview of defining exception conditions.

Chapter Contents

How to Format Exception Control Statements

Overview

This unit contains information on formatting exception control statements.

Use of exception control statements

You can use exception control statements to specify exception conditions for a batch job and automate the execution of common DB2 utilities. Each exception control statement defines a single exception condition.

How !DB/SMU processes exception control statements

You can place exception control statements directly into a batch job stream immediately following the EXCPTCTL DD statement, or you can create an ISPF member to contain them. If you choose the latter approach, you include the name of the member on the EXCPTCTL DD statement as shown in this example:

```
//EXCPTCTL DD DISP=SHR, DSN=&CNTL.(membername)
```

EXCPTCTL DD statements override profile exception conditions

When you provide exception control statements following the EXCPTCTL DD statement or in a control member, these exception conditions override any exception conditions you included in the exception profile you used for the job.

Conditions you can use in exception control statements

The valid exception conditions !DB/SMU provides for monitoring your DB2 system are in the next unit, “Exception Conditions for a Job Stream or Exception Control Member” on page 534.

Formatting guidelines for exception control statements

Apply the following guidelines when you format exception control statements in a batch job stream or an exception control member:

Guideline	Example
Use the correct format for each statement.	`condition-operator-value action`
Do not insert blanks between the condition and the relational operator, or between the relational operator and the value.	`ROWS>=1000000`
You can insert any number of blanks between the value and the action.	`ROWS>=1000000 REORG` or `ROWS>=1000000 REORG`

Valid relational operators you can use in exception control statements

This chart lists the valid relational operators you can use in an exception control statement.

Relational Operator	Meaning
=	Equals the specified value
¬=	Is not equal to the specified value
<	Is less than the specified value
>	Is greater than the specified value
<=	Is less than or equal to the specified value
>=	Is greater than or equal to the specified value

Valid values for triggering exception control statements

The value you specify is a threshold used to trigger an exception or a value to control the type of processing that !DB/SMU is to perform. What is entered for the Value parameter in an exception condition depends on the type of exception condition you specify. This chart provides the details.

Type of Exception Condition	Value To Enter
EXIT condition	Name of a user exception exit
CLUSTERING condition	Whether an index is a clustering index. **Y** for a clustering index, **N** for a nonclustering index, or **D** for clustering index by default.
COMPRESSION condition	Whether compression is in effect. **Y** if compression is in effect, **N** if it is not.
Any percentage condition	An integer between 0–100
All other conditions	An integer between 0–99,999,999

Valid actions for exception control statements

This chart lists the valid actions you can specify on an exception control statement.

Action	Result
COPY	Performs a full image copy
INCRCOPY	Performs an incremental image copy
RECOVER	Recovers the space
REORG	Reorganizes the space
REORGIX	If the object is an index, reorganizes it. If the object is a table space, reorganizes the associated indexes.
REORGTS	If the object is a table space, reorganizes it. If the object is an index, reorganizes the associated table space.
RUNSTATS	Updates the statistics in the DB2 catalog
SKIP	Cancels the printing of the errors and exceptional conditions report

Reminder about processing exception conditions

To review information about how !DB/SMU processes exception conditions and the order of significance of exception conditions, see "Order of Processing for Exception Conditions" on page 514.

Example exception control statements

This chart shows an example of correctly formatted exception control statements.

Purpose	Example
Set the following trigger values: • Make a full image copy if more than 10% of the pages contain updates. • Reorganize the table space if there are any pointer records. • Do not print the error and exception report if the table space occupies 24 pages or fewer.	**PCTPAGESUPDATED>10 FULLCOPY** **POINTERRECORDS>0 REORG** **PAGES<=24 SKIP**

A note about reorganizing table spaces based on index conditions

There are circumstances where you may want to reorganize table spaces based on a triggering condition you specify for indexes. An example might be, "reorganize those table spaces for which the associated index has a CLUSTERRATIO<80". It is not possible to perform this processing using inline exception control statements. Rather, you must write an exception exit, exiting with return code 28 (or 128), to perform this processing. You can specify the name of the exception exit you write on the EXCPTCTL DD statement. Information about writing exception exits is found in "Using Exception Exits" on page 541.

Exception Conditions for a Job Stream or Exception Control Member

Overview

This unit lists the valid exception conditions that you can add to a batch job stream or include in an exception control member.

Reminder about using exception conditions

Combine the exception conditions in this unit with valid relational operators and optional actions. See "How to Format Exception Control Statements" on page 530 for information on formatting and a list of valid operators and actions.

Exception conditions for both table spaces and index spaces

This chart lists the exception conditions you can use when defining an exception that will apply either to a table space or index space. For each exception condition, the chart lists the type of value you can apply.

Exception Condition	Value to Use
EXIT	Name of a user-defined exit
COPYFACTOR	Value (0–100) for the copy factor
EMPTYPAGES	Number of empty formatted pages
ERRORPAGES	Number of pages flagged by the scan as having data integrity errors
EXTENTS	Number of DASD extents in the table space or index space
PAGES	Number of pages in the table space or index space
PAGESAFTERREORG	Number of pages required after reorganizing this table space or index space
PAGESOCCUPIED	Number of pages that are in use
TRACKS	Number of tracks in the table space or index space
UNFORMATTEDPAGES	Number of pages in the table space or index space that have never been formatted by DB2

Exception conditions for table spaces

This chart lists the exception conditions you can use when defining an exception that applies to a table space. For each exception condition, the chart lists the type of value you can apply.

Exception Condition	Value to Use
AVGLENPERROW	Average row size in the table space
COMPRESSION	Whether or not a table space is compressed; enter Y for Yes and N for No
DAYSSINCEIMAGECOPY	Number of days since the last, full image copy
DROPPEDROWS	Number of dropped rows in the table space
PAGESSAVED%	Percentage (0–100) of the total space saved by compressing the table space
PCTPAGESUPDATED	Percentage (0–100) of pages modified since the previous image copy
ROWS	Number of rows in the table space
ROWSCOMPRESSED%	Percentage (0–100) of compressed rows in the table space
ROWSPERPAGE	Average number of rows per occupied page
TOTALDICTPACES	Number of pages used by the compression dictionary
TOTALROWSCOMPRESSED	Total number of compressed rows in the table space
UPDATEDEMPTYPAGES	Number of empty pages modified by the DELETE command
UPDATEDPAGES	Number of pages modified since the previous image copy
UPDATEDTRACKS	Number of tracks modified since the previous image copy
WASTEDDICTPAGES	Total number of pages wasted as a result of compressing a table space (This can occur in a segmented space.)

Exception conditions for pointer records

This chart lists the exception conditions you can use when defining an exception that applies to pointer records. For each exception condition, the chart lists the type of value you can apply.

Exception Condition	Value to Use
AVGPAGESPTRREC	Average number of pages between the pointer record and overflow page
FAROFFROWS	Number of overflow rows more than 1 track away from the pointer record
NEAROFFROWS	Number of overflow rows within 1 track of the pointer record
OVERFLOWRECORDS	Number of overflow records in the table space
POINTERRECORDS	Number of pointer records in the table space
REORGFACTOR	Value (0–100) for the reorg factor

Exception conditions for monitoring DASD use

This chart lists the exception conditions you can use when defining an exception for monitoring DASD use. For each exception condition, the chart lists the type of value you can apply.

Exception Condition	Value
FREESPACE	Amount of total space (in K bytes) that contains free space and unformatted pages
FREESPACE:&pct.	Percentage (0–100) of space that contains free space and unformatted pages

Exception conditions for monitoring DASD use (continued)

Exception Condition	Value
OVERHEAD	Amount of table space (in K bytes) that contains overhead
OVERHEAD%	Percentage (0–100) of total table space that contains overhead
PAGESWITHAVGROOM	Number of pages in the table space with room for an average size row
PAGESWITHLONGROOM	Number of pages in the table space with room for any size row
PAGESWITHNOROOM	Number of full pages in the table space
PAGESWITHSHORTROOM	Number of pages in the table space with room for only a shortest size row
ROWSDATA	Amount of space (in K bytes) that is occupied by table rows
ROWSDATA%	Percentage (0–100) of total space occupied by table rows
TOTALSPACE	Amount of total DASD space (in K bytes) allocated to the dataset
UNFORMATTEDSPACE	Amount of total table space and index space (in K bytes) that contains unformatted pages
UNFORMATTEDSPACE%	Percentage (0–100) of total space that is occupied by unformatted pages
WASTEDSPACE	Amount of table space (in K bytes) that is wasted space
WASTEDSPACE%	Percentage (0–100) of total table space that is wasted

Note: The spelling of the condition used to specify percentage of wasted space, when used in an exception control statement, differs from the same condition used in an exception profile. Refer to the chart for details.

Exception Control Statement	Exception Profile
WASTEDSPACE%	WASTEDSPACEPCT

Valid exception conditions for index spaces

This chart lists the exception conditions you can use when defining an exception for index spaces. For each exception condition, the chart lists the type of value you can apply.

Exception Condition	Value to Use
AVGDISTINPAGES	Average distance between leaf pages (measured in pages)
AVGDISTINTRACKS	Average distance between leaf pages (measured in tracks)
AVGKEYSPERSUBPAGE	Average number of index keys per subpage
AVGRECORDSPERKEY	Average number of records indexed per index key
AVGRECORDSPERPAGE	Average number of records per index page
AVGRECSPERSUBPAGE	Average number of records indexed per subpage
CLUSTERRATIO	Degree to which the physical ordering of rows in the table space matches the logical ordering of keys in the index
DISTINCTKEYS	Number of distinct key values
EMPTYSUBPAGES	Number of empty subpages in the index
KEYLENGTH	Length of the index key
LEAFPAGES	Number of leaf pages in the index

Valid exception conditions for index spaces (continued)

Exception Condition	Value to Use
LEVELS	Number of levels in the index
LEVELSAFTERREORG	Number of levels required after reorganizing the index space
PCTLEAFFREESPACE	Percentage (0–100) of space in the leaf pages that is free space
PCTPAGESUSED	Percentage (0–100) of the index pages that are being used
RECORDSINDEXED	Number of table records that are indexed
TOPRECORDSPERKEY	Highest number of table records that have the same key
TOTALINDEXPAGES	Total number of allocated pages in the index space
TOTALSUBPAGES	Total number of subpages in the index space
USEDINDEXPAGES	Number of leaf pages, intermediate pages, root pages, and DB2 control pages
USEDSUBPAGES	Total number of used subpages in the index space

Chapter 29.
Using Exception Exits

Introduction

Using exception exits in your batch jobs can help you automate the management of your DB2 system. This chapter helps you to define exception exits to assist in controlling !DB/SMU batch jobs.

Before using this chapter, read "Introduction to Batch Operations for !DB/SMU" on page 455 to gain an understanding of !DB/SMU batch functions. Also, read "Introduction to Handling Exception Conditions in Batch Jobs" on page 509 for an overview of defining exception conditions.

Chapter Contents

Background on Using Exception Exits

Overview

You can use CLIST or REXX programming to write an exception exit. This unit contains background on the uses of these exits.

Advantages of using an exception exit

Exception exits offer more processing options for monitoring your system and initiating actions than do exception profiles or exception control statements.

Using an exception exit provides the following advantages:

- It can automate detailed, conditional processing in batch operations.
- It provides a level of processing that far exceeds the capabilities of the keyword-only criteria specified in exception control statements or control statements.
- It provides a level of control that lets you vary the action to include a combination of conditions and identities of the spaces involved.

Examples of uses for exception exits

You can program options for detailed processing in an exception exit. For example, a user-written exception exit can do the following:

- Delegate logic to application-specific exits. These dependent exits can
  - Receive all the possible exception conditions from the first exit
  - Receive an appropriate subset of the exception conditions from the first exit
  - Be independent of the exception conditions
- Invoke the DB2 RUN process to update your choice of DB2 tables.
- Inform both the operator and the owner of any table space data integrity errors and invoke the DB2 RUN process to stop the table space.
- Invoke an exit to complete the following series of tasks:
  - Take action on combinations of conditions
  - Call another CLIST, REXX EXEC, or special program
  - Submit a job
  - Send a message
  - Read or write a file

How !DB/SMU passes values to exception exits

In addition to the keywords that represent exception conditions, you must use the keywords in this chart to pass information to !DB/SMU exception exits. Examples of the use of these keywords and the keywords that represent exception conditions can be found in the two example exception exits at the end of this chapter.

Keyword	Explanation
DBNAME	Database name
IXNAME	Index name
IXSPACE	Index space name
PARTITION	Partition number
TBCREATOR	Table AUTHID
TBNAME	Table name
TSNAME	Table space name

How !DB/SMU processes exception exits

The following considerations apply when !DB/SMU processes an exception exit.

- An exception exit supersedes all other actions specified in the job—for example, by means of an exception profile, exception control statements, or an exception control member.
- Any actions specified for other conditions are ignored if conditions invoke the exit.
- An exception exit is always invoked if you define a trigger condition, for example, PAGES>=0, that is always true.
- An exception exit is not invoked if the trigger condition is not true.

Example exception exits

The *hilev*.CLIST library contains example exception exits. They contain no logic but indicate programming requirements and the identity and applicability of each keyword.

Member Name	Type of Exit
KTSSMUXC	Sample CLIST
KTSSMURX	Sample REXX EXEC

Exit Codes for an Exception Exit

Overview

This unit contains describes exit codes for an exception exit.

Purpose of !DB/SMU exit codes

Use exit codes in your exception exit to provide the information !DB/SMU requires to initiate specific actions and invoke DB2 utilities.

Avoiding confusion between TSO return codes and !DB/SMU exit codes

!DB/SMU exit codes and TSO return codes can have the same numeric value. For example:

- The value *4* can represent a !DB/SMU exit code meaning “skip printing the error and exception report” or a TSO return code meaning “exception conditions were found”.
- The value *8* can represent a !DB/SMU exit code meaning “create an image copy” or a TSO return code meaning “a syntax error was encountered in the exception exit”.

To eliminate the confusion between !DB/SMU exit codes and TSO return codes, you can optionally add the value *100* to a !DB/SMU exit code. !DB/SMU recognizes and acts on either the initial value or the initial value plus 100. Thus, for example, you can:

- Use either 0 or 100 to print the error and exception report.
- Use either 16 or 116 to reorganize a table space.

Valid exit codes

The following chart lists the valid !DB/SMU exit codes.

Exit Code	Action for !DB/SMU to Take
0 or 100	Print the error and exception report; do not run any DB2 utilities.
4 or 104	Skip the error and exception report; do not run any DB2 utilities.
8 or 108	Perform a full image copy.
12 or 112	Perform an incremental image copy.
16 or 116	Run the DB2 REORG utility.
20 or 120	Run the DB2 RECOVERY utility.
24 or 124	Run the DB2 RUNSTATS utiility.
28 or 128	Reorganize the table space based upon index conditions.
32 or 132	Quiesce the space.
34 or 134	If the object is an index, reorganize it. If the object is a table space, reorganize the associated indexes.
36 or 136	If the object is a table space, reorganize it. If the object is an index, reorganize the associated table space.

Invoking an Exception Exit

Overview

This unit contains instructions for invoking an exception exit.

Exception situations where you can invoke an exception exit

You can use an exception profile, an exception control member, or exception control statements to call an exception exit.

Invoking an exception exit from an exception profile

Use these procedures to invoke an exception exit from an exception profile.

Step	Action
1	On the Primary Menu, type **5.1** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the Batch Run Profile Select panel.
3	To select the option for scan profiles, type **3** in the Select Profile Type field.
4	Press Enter. **Result**: !DB/SMU displays the Scan Profiles panel that lists the existing scan profiles.
5	To identify a profile you want to use, type **U** in the Sel field next to its name.
6	Press Enter. **Result**: !DB/SMU displays the Generate Batch Job panel.
7	Change or add the values as needed on the Generate Batch Job panel.
8	To identify the exception profile that specifies the exit you want to call, do one of the following: • Type the name of the profile in the Exception Profile field and press Enter. • Type **?** in the Exception Profile field and press Enter to display a list of profiles. Type **X** next to the name of the profile you want to use and press Enter. **Result**: !DB/SMU generates the JCL for the job and displays the Process Member panel.

Invoking an exception exit from an exception profile (continued)

Step	Action
9	On the Process Member panel, select the UPDATE NOW option. Type **1** in the Select Action field.
10	Press Enter. **Result**: !DB/SMU submits the batch job.

Invoking an exception exit from a batch job stream

Use the following procedures to invoke an exceptions exit from a batch job stream.

Step	Action
1	On the Primary Menu, type **5.1.1** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the Batch Run Profile Select panel.
3	To select the type of run profile you want to use, type its selection number in the Select Profile Type field.
4	Press Enter. **Result**: !DB/SMU displays a profiles panel that lists the existing run profiles for your selection.
5	To identify a profile you want to use, type **U** in the Sel field next to its name.
6	Press Enter. **Result**: !DB/SMU displays the Generate Batch Job panel.
7	Change or add the values as needed on the Generate Batch Job panel.
8	Press Enter. **Result**: !DB/SMU generates the JCL for the job and displays the Process Member panel.
9	On the Process Member panel, select the EDIT option by typing] **2** in the Select Action field.
10	Press Enter. **Result**: The member opens in the ISPF editor.

Invoking an exception exit from a batch job stream (continued)

Step	Action

11	Scroll to the //EXCPTCTL DD * statement. Add exception control statements to call the exit.
	Example: These statements invoke the exit if 10% or more of the pages have been updated.
	`//EXCPTCTL DD *`
	`EXIT=YOSEMITE`
	`PCTPAGESUPDATED>10`

12	Type **END** on the command line or press the appropriate function key. **Result**: !DB/SMU returns to the Process Member panel.
13	On the Process Member panel, select the UPDATE NOW option by typing **1** in the Select Action field.
14	Press Enter. **Result**: !DB/SMU submits the job.

Invoking an exception exit from an exception control member

Use the following procedures to invoke an exception exit from an exception control member.

Step	Action
1	Perform steps 1 through 9 as documented in the preceding section, “Invoking an exception exit from a batch job stream” on page 547.
2	Scroll to the //EXCPTCTL DD statement. Replace the member name EXCPTCTL (found in parentheses at the end of the statement) with the name of your exception control member that calls the exception exit. **Example**: This statement references a member named HOUSTON. `//EXCPTCTL DD DISP=SHR,DSN=&CNTL.(HOUSTON)`
3	Type **END** on the command line or press the appropriate function key. **Result**: !DB/SMU returns to the Process Member panel.

Invoking an exception exit from an exception control member (continued)

Step	Action
4	On the Process Member panel, select the UPDATE NOW option by typing **1** in the Select Action field.
5	Press Enter. **Result**: !DB/SMU submits the job.

Example CLIST for an Exception Exit

Overview

This unit contains an example CLIST that performs an exception exit.

Contents of a sample CLIST for an exception exit

The following example CLIST is KTSSMUXC, which is provided with !DB/SMU in the *hilev*.CLIST library. This example shows how a CLIST can implement an exception exit. It indicates the identity, nature, and applicability of each keyword.

```
PROC 0  DBNAME()          +
        TSNAME()          +
        IXNAME()          +
        ERRORPAGES()      +
        ROWS()            +
        PAGES()           +
        UPDATEDPAGES()    +
        POINTERRECORDS()  +
/*              ADDITIONAL PROC STATEMENTS COMPLETE THIS SAMPLE LIST     */

CONTROL NOLIST

/*****************************************************************************/
/*                                                                           */
/*              IF THE SPACE HAS ERRORS, INFORM THE OPERATOR                 */
/*                                                                           */
/*****************************************************************************/

IF  &ERRORPAGES NE 0  +
  THEN DO                             /*  THERE ARE ERRORS                */
    IF  &STR(&IXNAME) = &STR()  +
      THEN DO                         /*  THIS IS A TABLE SPACE           */
        SE 'INTEGRITY ERROR IN TABLE SPACE &TSNAME..' CN(1)
      END
      ELSE DO                         /*  THIS IS AN INDEX SPACE          */
        SE 'INTEGRITY ERROR IN INDEX SPACE &IXNAME..' CN(1)
      END
    EXIT CODE(0)                      /*  TELL DB2-SMU TO PRINT REPORTS   */
  END

/*****************************************************************************/
/*                                                                           */
/*                         SET DEFAULT REPORTING                             */
/*                                                                           */
/*****************************************************************************/

SET &PRINT = &STR(YES)                /*  BY DEFAULT, PRINT REPORTS       */
IF  &PAGES <= 24  +
  THEN DO                             /*  NO LARGER THAN 2 3390 TRACKS    */
    SET &PRINT = &STR(NO)             /*  DON'T PRINT REPORTS             */
  END
```

Contents of a sample CLIST for an exception exit (continued)

```
/*********************************************************************/
/*                                                                   */
/*                  PROCESS SPACES IN TEST DATA BASES                */
/*                                                                   */
/*********************************************************************/

SET &L = &LENGTH(&STR(&DBNAME))
IF  &L >= &LENGTH(&STR(TEST))  THEN  SET &L = &LENGTH(&STR(TEST))
IF  &SUBSTR(1:&L,&STR(&DBNAME)) = &STR(TEST)  +
  THEN DO                           /*  THIS IS A TEST DATA BASE      */
    EXIT CODE(4)                    /*  TELL DB2-SMU TO SKIP REPORTS  */
  END
/*********************************************************************/
/*                    SPACE IS NOT IN A TEST DATA BASE               */
/*********************************************************************/

IF  &STR(&IXNAME) = &STR()  +
  THEN DO                           /*  THIS IS A TABLE SPACE         */

/*********************************************************************/
/*                        TABLE SPACE PROCESSING                     */
/*********************************************************************/

    SET &REORG = &STR(NO)           /*  BY DEFAULT, DON'T REORG       */
    IF  &POINTERRECORDS * 1000 > &ROWS  +
    THEN DO                         /*  POINTERS EXCEED 0.1% ROWS     */
      SET &REORG = &STR(YES)        /*  REORG                         */
    END

    SET &COPY  = &STR(NO)           /*  BY DEFAULT, DON'T COPY        */
    IF  &UPDATEDPAGES > 0  +
      THEN DO                       /* THERE ARE UPDATES              */
        SET &COPY = &STR(PART)      /*  PARTIAL COPY                 */
        IF  &PAGES <= 900  OR  &PCTPAGES >= 667  +
          THEN DO                   /*  SIZE IS OVER 5 CYLS OF 3390  */
                                    /*  OR 2/3 OF SPACE IS UPDATED   */
            SET &COPY = &STR(FULL)  /*  FULL COPY                    */
          END
        END

    IF  &STR(&REORG) = &STR(YES)   +
      THEN DO
        EXIT CODE(16)               /*  TELL DB2-SMU TO REORG         */
      END
    IF  &STR(&COPY)  = &STR(FULL)  +
      THEN DO
        EXIT CODE(12)               /*  TELL DB2-SMU TO FULL COPY     */
      END
    IF  &STR(&COPY)  = &STR(PART)  +
      THEN DO
        EXIT CODE(8)                /*  TELL DB2-SMU TO PARTIAL COPY  */
      END
    IF  &STR(&PRINT) = &STR(NO)    +
      THEN DO
        EXIT CODE(4)                /*  TELL DB2-SMU TO SKIP REPORTS  */
      END
    EXIT CODE(0)                    /*  TELL DB2-SMU TO PRINT REPORTS */
  END
```

Contents of a sample CLIST for an exception exit (continued)

```
  ELSE DO                              /*  THIS IS AN INDEX SPACE        */

/*****************************************************************************/
/*                                                                           */
/*                          INDEX SPACE PROCESSING                           */
/*                                                                           */
/*****************************************************************************/

    /***  NO SPECIFIC ACTION IN THIS EXAMPLE  ***/

    IF  &STR(&PRINT) = &STR(NO)    +
      THEN DO
        EXIT CODE(4)                   /*  TELL DB2-SMU TO SKIP REPORTS   */
      END
    EXIT CODE(0)                       /*  TELL DB2-SMU TO PRINT REPORTS  */
  END
```

A reminder about exit codes

In a CLIST, always place parentheses around any exit codes for !DB/SMU.

Right	Wrong
`EXIT CODE(0)`	`EXIT CODE 0`

Example REXX EXEC for an Exception Exit

Overview

This unit contains an example REXX EXEC that can be called for an exception exit.

Sample contents a REXX EXEC for an exception exit

The following is the sample REXX EXEC called KTSSMURX, which is provided with !DB/SMU in the *hilev*.CLIST library. This example shows how a REXX EXEC can implement an exception exit. It indicates the identity, nature, and applicability of each keyword.

```
/* REXX */

/*********************************************************************/
/*                                                                   */
/*               define REXX variables and assign values             */
/*                                                                   */
/*********************************************************************/

parse arg a
do i = 1 to words(a)
  parse value word(a,i) with kw "(" value ")" .
  interpret kw "= '"value"'"
  end

/*********************************************************************/
/*                                                                   */
/*              if the space has errors, inform the operator         */
/*                                                                   */
/*********************************************************************/

if  errorpages ¬= 0
  then do                             /*  there are errors           */
    if  symbol('ixname') ¬= 'VAR'
      then do                         /*  this is a table space      */
        "SE 'Integrity error in table space "tsname".' CN(01)"
      end
      else do                         /*  this is an index space     */
        "SE 'Integrity error in index space "ixname".' CN(01)"
      end
    exit 0                            /*  tell  DB/SMU to print reports  */
  end

/*********************************************************************/
/*                                                                   */
/*                         set default reporting                     */
/*                                                                   */
/*********************************************************************/

print = "yes"                         /*  by default, print reports  */
if  pages <= 24
  then do                             /*  no larger than 2 3390 tracks  */
    print = "no"                      /*  don't print reports        */
  end
```

Sample contents a REXX EXEC for an exception exit (continued)

```
/*****************************************************************************/
/*                                                                           */
/*                    process spaces in test data bases                      */
/*                                                                           */
/*****************************************************************************/

if  left(dbname,length(test)) = test
  then do                              /*  this is a test data base       */
    exit 4                             /*  tell  DB/SMU to skip reports   */
  end
/*****************************************************************************/
/*                                                                           */
/*                     space is not in a test data base                      */
/*                                                                           */
/*****************************************************************************/

if  symbol('ixname') ¬= 'VAR'
  then do                              /*  this is a table space          */

/*****************************************************************************/
/*                                                                           */
/*                          table space processing                           */
/*                                                                           */
/*****************************************************************************/

    reorg = "no"                       /*  by default, don't reorg        */
    if  pointerrecords * 1000 > rows
    then do                            /*  pointers exceed 0.1% rows      */
      reorg = "yes"                    /*  reorg                          */
    end

    copy = "no"                        /*  by default, don't copy         */
    if  updatedpages > 0
      then do                          /* there are updates               */
        copy = "part"                  /*  partial copy                   */
        if  pages <= 900  |  pctpages >= 667
          then do                      /*  size is over 5 cyls of 3390    */
                                       /*  or 2/3 of space is updated     */
            copy = "full"              /*  full copy                      */
          end
        end

    if  reorg = "yes"
      then do
        exit 16                        /*  tell  DB/SMU to reorg          */
      end
    if  copy  = "full"
      then do
        exit 12                        /*  tell  DB/SMU to full copy      */
      end
    if  copy  = "part"
      then do
        exit 8                         /*  tell  DB/SMU to partial copy   */
      end
    if  print = "no"
      then do
        exit 4                         /*  tell  DB/SMU to skip reports   */
      end
    exit 0                             /*  tell  DB/SMU to print reports  */
  end
```

Sample contents a REXX EXEC for an exception exit (continued)

```
  else do                              /*  this is an index space         */

/*****************************************************************************/
/*                                                                           */
/*                          index space processing                           */
/*                                                                           */
/*****************************************************************************/

    /***  no specific action in this example  ***/

    if  print = "no"
      then do
        exit 4                         /*  tell  DB/SMU to skip reports   */
      end
    exit 0                             /*  tell  DB/SMU to print reports  */
  end
```

Chapter 30. Specifying Actions Using Action Profiles and Inline User Action Definitions

Introduction

You can cause !DB/SMU to automatically take an action or series of actions as the result of triggers you identify using exception conditions. The actions !DB/SMU takes can be predefined in an action profile. Alternatively, the actions can be specific to a given !DB/SMU batch job when included inline in the !DB/SMU batch JCL as a user action definition.

Using action profiles in your batch jobs can help you automate the management of your DB2 system. This chapter provides you with an overview of creating action profiles as well as creating and making available to !DB/SMU macros comprising user-defined actions.

Before using this chapter, read "Introduction to Batch Operations for !DB/SMU" on page 455 to gain an understanding of !DB/SMU batch functions. If you need to review information about exception conditions, see "Introduction to Handling Exception Conditions in Batch Jobs" on page 509.

Chapter Contents

Specifying Actions for a !DB/SMU Batch Execution

Overview

This unit introduces you to specifying actions !DB/SMU can take during a batch run.

Types of actions you can specify

The types of actions !DB/SMU can take during a batch execution are described below. Each of these types of actions are described in detail in subsequent units in this chapter.

- !DB/SMU initiates an action profile and executes the steps it contains. You specify the name of the action profile to use when you fill out the Generate Batch Job panel for a scan run profile.
- !DB/SMU executes the steps contained in an inline user action definition that you name on the EXCPTCTL DD statement. The valid names or *labels* that you can use on the EXCPTCTL DD statement are described later in this chapter.

  If the label you specify on the EXCPTCTL DD statement does not have a corresponding action macro in-line in the batch JCL, !DB/SMU executes the DB2 utility that the label refers to.

Information Sources

Refer to this chart for information about the various types of actions !DB/SMU can take.

Type of Action	Source of Information
Action profile	pp.559 through 574 in this chapter.
User action definition	pp.576 through 589 in this chapter.

Introduction to Using Action Profiles

Overview

This unit presents an overview of using action profiles.

What is an action profile?

An action profile lets you predefine the instructions needed to perform a sequence of job steps, including the invocation of one or more DB2 utilities. You include an action profile in your batch job by naming it on the Generate Batch Job panel. !DB/SMU executes the steps it contains when the action profile is triggered during a batch run as the result of exception processing.

Benefits from using action profiles

Using an action profile permits you to gain these benefits:

- simplifies the process of creating and submitting batch jobs to manage DB2 through:
  - a structured interface to easily define and maintain action profiles
  - extensive online help for defining action conditions
  - a convenient means to save and apply frequently used action conditions
- reduces errors by automatically generating the job stream for complex actions
- permits the initiation of multiple DB2 utilities within one job
- permits the tailoring of job operations with special job control statements

Using an action profile to run multiple DB2 utilities

Using an action profile, you can execute a single batch job to invoke multiple DB2 (or OEM) utilities in any sequence that suits your needs. You can initiate these DB2 utilities and procedures:

- COPY (for full or incremental copies)
- MERGECOPY
- MODIFY
- QUIESCE
- REBIND (for plans and packages)
- RECOVERY
- REORG
- RUNSTATS

Relationship of an action profile to other profile types

A batch job that includes an action profile requires that these profile types also be included: either a scan run profile or a monitor run profile, and an exception profile. You name all three of these profile types when creating a batch job using the !DB/SMU Generate Batch Job panel. This chart explains the function of each profile type:

Type of Profile	Function
Scan or monitor run profile	Identifies the spaces for the job
Exception profile	Specifies the exception conditions that trigger an action
Action profile	Specifies the sequence of job steps to be taken when the action profile is triggered

Dependencies for an action profile

An action profile is executed against only those spaces that are scanned by the associated scan run profile and meet the exception condition defined in the associated exception profile.

Accessing and Managing Existing Action Profiles

Overview

!DB/SMU provides easy access to existing action profiles. This unit contains instructions for accessing the Action Profiles panel. This scrollable panel lists existing action profiles and their descriptions.

Accessing existing action profiles

Follow these steps to access a list of existing action profiles.

Step	Action
1	On the Primary Menu, type **5.3** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the Action Profiles panel.

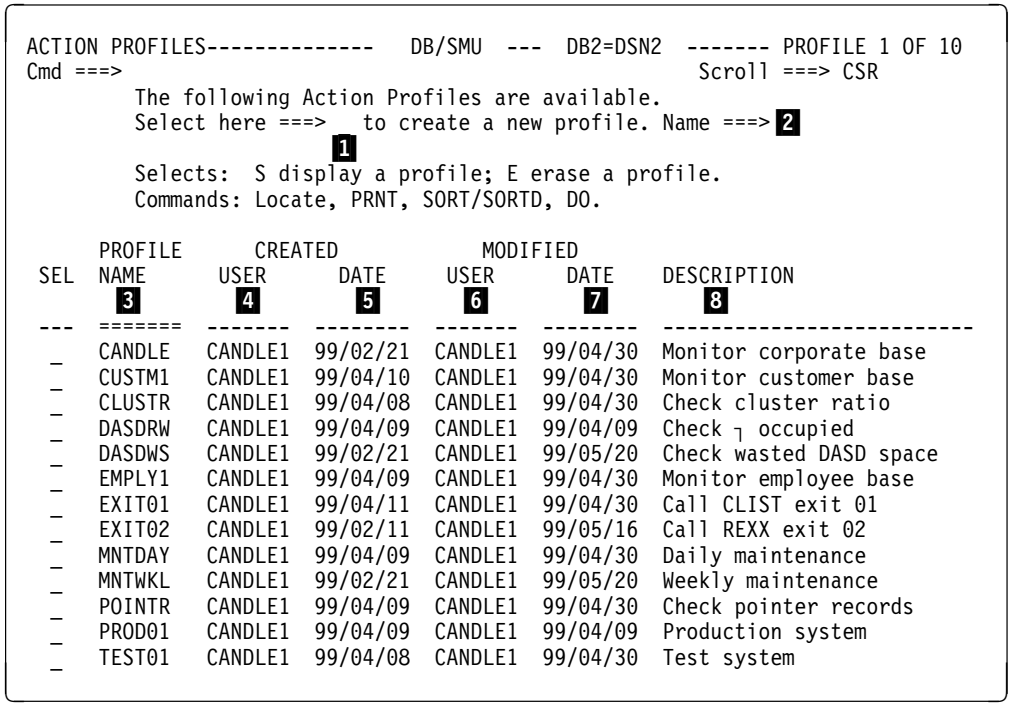
Managing existing action profiles

Use the Action Profiles panel to manage existing action profiles. Refer to the following chart for details.

Task	Procedure
Display the definitions in an existing action profile. (If you created the action profile, you can also modify it.)	Type **S** in the SEL field next to the name of the profile and press Enter. **Result**: !DB/SMU displays the Action Profile *profilename* panel.
Erase an existing action profile (only if you were its creator).	Type **E** in the SEL field next to the name of the profile and press Enter. **Result**: !DB/SMU erases the name of the profile from the list and deletes the member from the user PDS.

Elements of the Action Profiles panel

This is an example Action Profiles panel.



```
ACTION PROFILES--------------   DB/SMU  ---  DB2=DSN2  ------- PROFILE 1 OF 10
Cmd ===>                                                 Scroll ===> CSR
        The following Action Profiles are available.
        Select here ===>   to create a new profile. Name ===> 2
                        1
        Selects:  S display a profile; E erase a profile.
        Commands: Locate, PRNT, SORT/SORTD, DO.

     PROFILE      CREATED           MODIFIED
 SEL  NAME      USER      DATE     USER      DATE     DESCRIPTION
       3         4         5        6         7         8
 ---  =======  =======  ========  =======  ========  --------------------------
  _   CANDLE   CANDLE1  99/02/21  CANDLE1  99/04/30  Monitor corporate base
  _   CUSTM1   CANDLE1  99/04/10  CANDLE1  99/04/30  Monitor customer base
  _   CLUSTR   CANDLE1  99/04/08  CANDLE1  99/04/30  Check cluster ratio
  _   DASDRW   CANDLE1  99/04/09  CANDLE1  99/04/09  Check ┐ occupied
  _   DASDWS   CANDLE1  99/02/21  CANDLE1  99/05/20  Check wasted DASD space
  _   EMPLY1   CANDLE1  99/04/09  CANDLE1  99/04/30  Monitor employee base
  _   EXIT01   CANDLE1  99/04/11  CANDLE1  99/04/30  Call CLIST exit 01
  _   EXIT02   CANDLE1  99/02/11  CANDLE1  99/05/16  Call REXX exit 02
  _   MNTDAY   CANDLE1  99/04/09  CANDLE1  99/04/30  Daily maintenance
  _   MNTWKL   CANDLE1  99/02/21  CANDLE1  99/05/20  Weekly maintenance
  _   POINTR   CANDLE1  99/04/09  CANDLE1  99/04/30  Check pointer records
  _   PROD01   CANDLE1  99/04/09  CANDLE1  99/04/09  Production system
  _   TEST01   CANDLE1  99/04/08  CANDLE1  99/04/30  Test system
```

1 Position in which to type any character to indicate you want to create a new profile

2 Name for your new profile

3 Name of each existing action profile

4 User ID of the person creating each profile

5 Date that each profile was created

6 User ID of the person who last modified each profile

7 Date that each profile was last modified

8 Optional description you can provide for each profile

Sorting the list of action profiles

You can use the SORT and SORTD commands to sort the list in ascending or descending order on any column except DESCRIPTION.

Editing an action profile

You cannot edit an action profile with ISPF or another external editor. To edit an existing action profile, follow the steps in this chapter in the unit "Using the Custom Definition Panel to Expand Your Action Profile" on page 567.

Overriding actions in an action profile

Action statements included in the batch JCL following the EXCPTCTL DD statement take precedence over the actions in an action profile. To temporarily override the actions in an action profile, add the necessary action control statements to the generated job stream. Instructions for doing so are found in "Providing Inline User Action Definitions to Specify Actions" on page 576.

Overview of Defining an Action Profile

Overview

This unit contains the initial procedures for beginning a definition for an action profile.

Elements of the Action Profile profilename panel

This is the Action Profile *profilename* panel.

```
ACTION PROFILE UTIL01 1 --------   DB/SMU  ---  DB2=DSN2  ----------------------
Cmd ===>

Description: 2  Optionally enter a description of the profile here.

                     Select the action definition to be
                   3 Edited ===>       3 Deleted ===>

                     1. REORG.
                     2. REORG Table Space.
                     3. REORG Index Space.
                     4. Full Image Copy.
                     5. Partial Image Copy.
                   4 6. Recover.
                     7. Recover Table Space.
                     8. Recover Index Space.
                     9. RUNSTATS.
                     10. RUNSTATS Table Space.
                     11. RUNSTATS Index Space.

                  Highlighted entries exist in this profile.
```

1 Name you have given to this action profile

2 Optional description you supply for the action profile

3 Number of the action you want to perform—either adding a primary action to the profile or deleting it from the profile

4 Primary actions you can choose

Process for defining an action profile

Defining an action profile is a two-phase process. See the chart for information about each phase.

Phase	Source of Information
Phase 1—use the Action Profile *profilename* panel to identify the primary action you want the action profile to perform.	"Identifying the Primary Action for the Action Profile" on page 566
Phase 2—use a custom definition panel, displayed after you have selected a primary action on the Action Profile *profilename* panel, to add action statements. These action statements permit you to include the following in your profile definition: • Additional DB2 utilities • DB2 procedures • Job control actions	"Using the Custom Definition Panel to Expand Your Action Profile" on page 567

Identifying the Primary Action for the Action Profile

Overview

This unit describes the use of the Action Profile *profilename* panel to identify the primary action you want your action profile to perform.

Phase 1—identifying the primary action for the profile

Use the following procedures to create an action profile.

Step	Action
1	On the Primary Menu, type **5.3** in the Option field.
2	Press Enter. **Result**: !DB/SMU displays the Action Profiles panel.
3	Move the cursor to the Select here field and type **S** in the field.
4	Move the cursor to the Name field. Type a name for the new action profile in the field.
5	Press Enter. **Result**: !DB/SMU displays the Action Profile *profilename* panel. This panel lists the DB2 utilities you can include in an action profile.
6	Select a primary action to add to or delete from the profile you are defining: • To add an action definition to the action profile, type its number in the Edited field. • To delete an action definition from the action profile, type its number in the Deleted field.
7	Press Enter. **Result**: The result depends on whether you are adding or deleting an action definition: • If you want to add an action definition !DB/SMU displays a panel on which you can customize your definition. See "Using the Custom Definition Panel to Expand Your Action Profile" on page 567. • If you want to delete an action definition, !DB/SMU performs the delete.

Using the Custom Definition Panel to Expand Your Action Profile

Overview

This unit contains instructions for customizing your action profile.

Background about types of actions for custom definitions

!DB/SMU provides a custom definition panel for customizing operations for a utility. You can combine the following types of actions in a sequence that suits your needs:

- DB2 utilities like COPY, QUIESCE, or REORG
- DB2 REBIND command for plans and packages
- Job control statements like EXEC or COPY JOBSTREAM

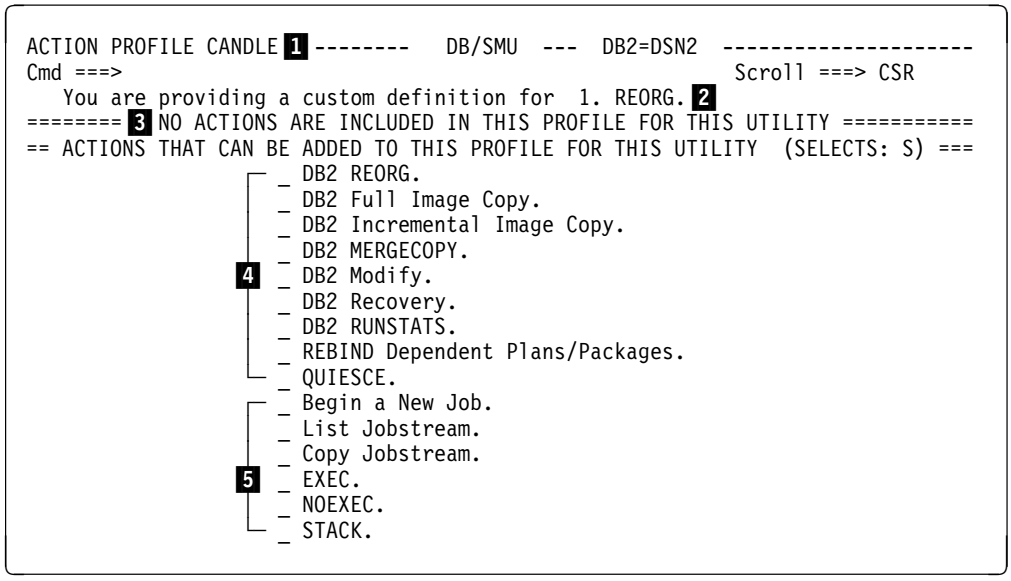
About the custom definition panel for an action profile

This panel has two parts:

- The upper part contains an ordered list of all the actions currently included in the custom definition for the utility.
- The lower part lists all the actions that you can add to the customized definition of this utility.

Elements of the custom definition panel

This is the custom definition panel, displayed when you select a primary action on the Action Profile *profilename* panel.



1 Name of the profile you are creating. It is the same as the name that appeared on the Action Profile *profilename* panel.

2 Information !DB/SMU provides to tell you the primary action for which you are designing a custom action profile.

3 If you are creating a new definition for a utility, there are no actions in the upper part of the panel. The message **NO ACTIONS ARE INCLUDED IN THIS PROFILE FOR THIS UTILITY** appears instead.

4 DB2 utility actions or processes you can include in your custom definition.

5 Job control actions you can include in your custom definition.

Phase 2—Customizing the operation of a utility in an action profile

Use the following procedures to customize a utility in an action profile.

Step	Action
1	On the Action Profile *profilename* panel, choose a utility action to edit and press Enter. **Result**: !DB/SMU displays the custom definition panel for the action profile.
2	Select and order actions for any DB2 utilities, commands, and job steps that you want to include in the operation of the utility. (See "Selecting processing control commands for an action profile on the custom definition panel" on page 572 for details.)

Selecting actions on the custom definition panel for an action profile

Refer to the following chart for instructions on making choices on the custom definition panel for an action profile.

Task	Procedure	Additional Information
Add new actions to the definition.	Place the cursor in the lower part of the panel next to the action you want to add. Type **S** in the field next to the action and press Enter. You can select multiple actions by typing **S** in multiple fields before you press Enter.	The new action moves to the bottom of the list in the upper part of the panel. When the profile is executed, each action is completed before the next one is started.
Copy an existing action.	Place the cursor next to the action you want to copy and type **C** in the field next to it. Type a destination select next to another action in the list and press Enter.	Inserts a duplicate action into the list of current actions at the point you specify. The destination selects are: • **B** to place an action before an entry • **A** to place an action after an entry

Selecting actions on the custom definition panel for an action profile (continued)

Task	Procedure	Additional Information
Delete an existing action.	Place the cursor next to the action you want to delete. Type **D** in the field next to the action and press Enter.	Deletes the action from the list of current actions.
Rearrange the sequence of included actions.	Type **M** next to the action you want to move. Type a destination select next to another action to relocate the action in the list and press Enter.	The destination selects are: • **B** to place an action before an entry • **A** to place an action after an entry

Considerations when selecting multiple types of spaces in an action profile

For some utilities, such as REORG, !DB/SMU provides options for applying the utility to

- Table spaces only
- Index spaces only
- Table spaces and index spaces

An action in an action profile can initiate a DB2 utility that

- Applies to both table spaces and index spaces
- Applies specifically either to table spaces or index spaces

If actions in an action profile apply to both types of spaces and other actions apply to a specific type of space, !DB/SMU gives precedence to the actions that apply to a specific type of space. If you later delete actions that apply to a specific type of space, then !DB/SMU executes the actions for both types of spaces.

Example when selecting multiple types of spaces in an action profile

This example shows how the actions initiated by the profile change:

Example of Selecting Multiple Utilities for an Action Profile	Result
You include actions for REORG Index Space and RUNSTATS in a profile.	The profile runs REORG and RUNSTATS against index spaces only
You delete the REORG Index Space action and leave the RUNSTATS action in the profile.	The profile now runs RUNSTATS against both tables spaces and index spaces.

Using Processing Control Commands in an Action Profile

Overview

This unit describes the processing control commands you can use in when creating a custom definition for an action profile.

Selecting processing control commands for an action profile on the custom definition panel

Processing control commands determine how !DB/SMU processes the generated job streams. The following chart explains how the processing control commands work together.

Command	Use
BEGIN A NEW JOB	Starts a new job and overrwrites any prior job stream that exists in the *userid*U member. If you have submitted the prior job, use this action to begin a new job. Do not use this action if you have deferred the submission of the job. Unless you use the BEGIN A NEW JOB command, each subsequent job stream is appended to the prior one. See "Recommendations for a sequence of job actions in an action profile" on page 575 in the next unit for recommendations related to this command.
COPY JOBSTREAM	Copies the job stream to a destination that you specify. Use this action to defer submission of the job by copying it to a job scheduler.
EXEC	Submits the job for each action as it is requested—that is, each action occurs as a separate job.
LIST JOBSTREAM	Issues the PDSLIST instruction to list the job stream when the EXEC action preceeds this action. If the NOEXEC action preceeds this action, the job is not listed.

Selecting processing control commands for an action profile on the custom definition panel (continued)

Command	Use
NOEXEC	Defers the submittal of the job and saves it in the *userid*U member in the user PDS. Use NOEXEC to avoid creating separate jobs. For more information, see "Recommendations for a sequence of job actions in an action profile" on page 575 in the next unit, "An Example Custom Definition for an Action Profile."
STACK Only 1 STACK command is valid in each action profile. If you use the STACK command, it must be the last statement in your action profile.	Determines the number of DB2 spaces passed to a given DB2 utility generation. STACK can be a value from 1-32,760. By default, STACK = 1, causing each DB2 space to be processed individually.

Specifying the value for the STACK action

If you include the STACK action in an action profile, !DB/SMU displays a pop-up window in which you specify the value for STACK. Acceptable values are 1–32760.

Detailed information about the use of the STACK command

There are several factors that influence the actions !DB/SMU takes as a result of the use of the STACK command. These factors are fully detailed in "Using the STACK Processing Control Command" on page 583.

An Example Custom Definition for an Action Profile

Overview

This unit provides an example custom definition you can use as a model for creating your own.

Example custom definition for a REORG

The following example shows the Action Profile *profilename* panel with a custom definition for a REORG. The example uses multiple jobs.

```
ACTION PROFILE CANDLE ----------   DB/SMU  ---  DB2=DSN2  ---------------------
Cmd ===>                                                    Scroll ===> CSR
   You are providing a custom definition for  1. REORG.
=== ACTIONS INCLUDED IN THIS PROFILE FOR THIS UTILITY  (SELECTS: A,B,C,D,M) ===
                        _ Begin a New Job.
                        _ NOEXEC.
                        _ DB2 Full Image Copy.
                        _ DB2 REORG.
                        _ DB2 Full Image Copy.
                        _ DB2 RUNSTATS.
                        _ REBIND Dependent Plans/Packages.
                        _ EXEC.
                        _ List Jobstream.
                        _ Copy Jobstream.
== ACTIONS THAT CAN BE ADDED TO THIS PROFILE FOR THIS UTILITY  (SELECTS: S) ===
                        _ DB2 REORG.
                        _ DB2 Full Image Copy.
                        _ DB2 Partial Image Copy.
                        _ DB2 MERGECOPY.
                        _ DB2 Modify.
                        _ DB2 Recovery.
                        _ DB2 RUNSTATS.
                        _ QUIESCE.
                        _ STACK.
```

Recommendations for a sequence of job actions in an action profile

When !DB/SMU generates JCL, the default action for that JCL is SUBMIT. This may create circumstances whereby you submit separate jobs when a single job is desired, or concatenate job actions inadvertently, or inadvertently create a recursive job. To avoid these conditions, follow the recommendations in this table.

Recommendation	Explanation
In general, begin each action definition with the BEGIN A NEW JOB and NOEXEC commands. Without these job commands, the first action in the new definition becomes part of the final action for the previous space.	For example, if a definition specifies actions A, B and C, and the prior contents of member *userid*U were X, then the final contents of this member becomes XABC. **Caution:** If the final action for X was submission of a utility job, and you neglect to include the BEGIN A NEW JOB command prior to specifying actions A, B, and C, you would be submitting a utility job that submits a utility job, ad infinitum.
Use the NOEXEC action to avoid submitting separate jobs.	If you omit NOEXEC, three jobs are submitted: XA, XAB and XABC.

Providing Inline User Action Definitions to Specify Actions

Overview

You can also create instructions for multiple actions and job steps by using exception control statements. This unit explains this alternative procedure, which performs the same function as an action profile.

Background

A user action definition includes statements that initiate the actions for the DB2 utilities and job processing. These instructions for EXCPTCTL processing become your own *user action definition* for the DB2 utility.

Guidelines for creating a user action definition

These guidelines apply to the syntax of a user action definition.

Guideline	Additional Information
!DB/SMU checks the syntax of a definition during initialization.	Anything within a definition that !DB/SMU does not understand terminates the run before it starts.
An user action definition begins with the name of the definition and ends either explicitly with the END command or implicitly with a new action definition	The names you *must* use for user action definitions are: • COPY • FULLCOPY • INCRCOPY • RECOVER • RECOVERIX • RECOVERTS • REORG • REORGIX • REORGTS • RUNSTATS • RUNSTATSIX • RUNSTATSTS

Guidelines for creating a user action definition (continued)

Guideline	Additional Information
A user action definition can contain only three types of statements.	Acceptable types of statements: • UTIL command statements for the DB2 utilities • Processing control commands that control the flow of the batch job • Comments statements, identified by an asterisk (*) in column 1

Using the UTIL Command in Your User Action Definition

Overview

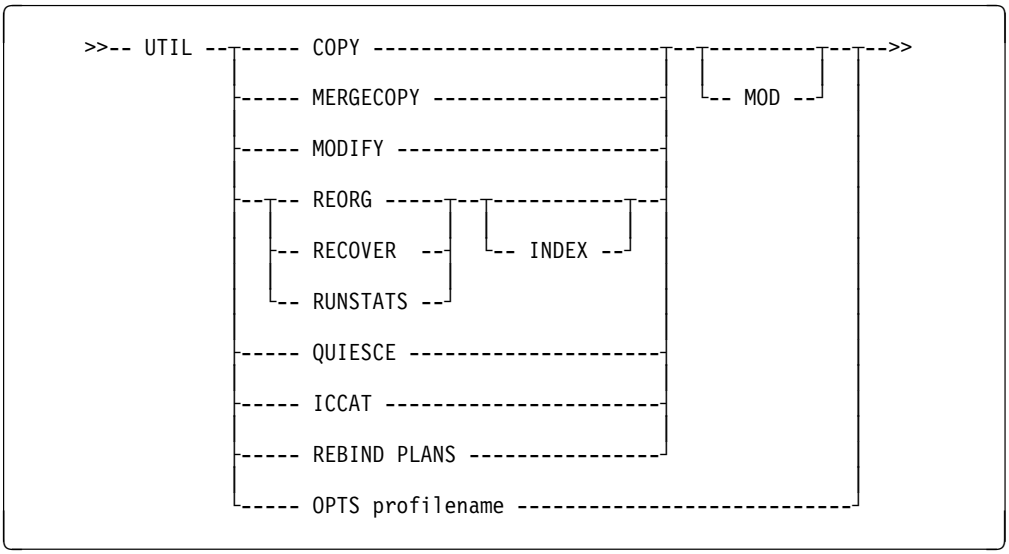
This unit describes the primary component of an inline user action definition—the UTIL command.

Purpose of the UTIL command

Use the UTIL command in your user action definition to specify the DB2 utility or process you want to execute.

Syntax of the UTIL command

This diagram provides the syntax for the UTIL command.



Explanation of the UTIL command syntax

This chart describes the syntax of the UTIL command.

Syntax Element for the UTIL Command	Explanation
COPY	Initiates the COPY utility to make full image copies.
MERGECOPY	Initiates the MERGECOPY utility to do one of the following: • Make one incremental copy from several incremental copies • Make a new, full copy from several incremental copies and a full copy
MODIFY	Initiates the MODIFY utility to remove records about recovery history from the DB2 catalog.
REORG	Reorganizes a table space. (When combined with the INDEX value, the reorganization is applied only to an index space.)
RECOVER	Recovers a table space. (When combined with the INDEX value, the reorganization is applied only to an index space.)
RUNSTATS	Initiates the RUNSTATS utility to update the statistics columns in the DB2 catalog. (When combined with the INDEX value, the statistics are updated only for index spaces.)

Explanation of the UTIL command syntax (continued)

Syntax Element for the UTIL Command	Explanation
QUIESCE	Initiates the QUIESCE utility to establish a quiesce point for one or more table spaces.
ICCAT	Initiates the SPCICCAT utility to do the following: • Speed the copying of multiple datasets to one tape by not rewinding the tape between steps. • Catalog image copies in the operating system catalog to avoid operational difficulties that can occur if the operating system cannot track these datasets.
REBIND PLANS	Rebinds all the plans and packages that are dependent on the space being processed.
OPTS *profilename*	Specifies the active DB2 utility profile. The profile must be in the system PDS.
MOD	MOD appends the output of an action to existing output for an action. Use MOD with any action *except* OPTS.

Using Processing Control Commands in a User Action Definition

Overview

This unit describes the commands you use inline in your user action definition to control the flow of processing.

Processing control commands

You can include processing control commands in a user action definition. The following chart contains their descriptions.

Command	Use
END	Ends the job stream
EXEC	Causes the immediate submission of a job once !DB/SMU has generated the JCL and control statements. (EXEC is the default.)
NOEXEC	Prevents the immediate submission of a job once !DB/SMU has generated the JCL and control statements. Use NOEXEC as the first command in the definition and EXEC as the last command if the job contains multiple operations for the same space.
PDSCOPY	Copies the generated JCL and control statements to a user-specified destination. Use PDSCOPY when you want to use a job scheduling package to initiate a job rather than !DB/SMU initiating a job when it is generated.
PDSEDIT	Applies an ISPF edit macro to the results of a batch job. For example, you can use PDSEDIT to set job-accounting data to be consistent with the job name.

Processing control commands (continued)

Command	Use
PDSEXEC	Executes the PDS member.
PDSLIST	Lists the JCL and control statements.
STACK=*nn*	Controls how many objects in a given queue are passed to the action profile or user action definition JCL.

Detailed information about the use of the STACK command

There are several factors that influence the actions !DB/SMU takes as a result of the use of the STACK command. These factors are fully detailed in "Using the STACK Processing Control Command" on page 583.

Using the STACK Processing Control Command

Overview

This unit discusses some of the specific conditions applicable to the use of the STACK control command.

Background about the use of the STACK control command

During an execution of the batch utility, !DB/SMU builds queues of objects (table spaces, index spaces) that meet the exception conditions you define. Satisfying the exception condition triggers the generation of the JCL to execute the appropriate action profile or user action definition. The number you specify for the STACK command controls how many objects in a given queue are passed to the the action profile or user action definition JCL.

Default for the STACK command

For a scan run, the default for STACK is 1. If STACK is not specified, the objects are passed 1 object at a time.

For a monitor run, if STACK is not specified, then all objects are passed to the generator as a single group.

When to use a STACK command

The STACK command can be used as follows:

- As a command you manually enter into your job stream
- In an action profile
- In a user action definition

Refer to the chart for information about using STACK in any of these circumstances.

How STACK is Used	Source of Additional Information
As a manually-entered command	“Commands for Batch Jobs” on page 482
In an action profile	“Using Processing Control Commands in an Action Profile” on page 572
In an user action definition	“Using Processing Control Commands in a User Action Definition” on page 581

Position of the STACK command in a job

STACK must always be the last command in your action profile definition or user action definition. When including STACK in a user action definition, the command must begin in column 1 of your JCL.

If STACK is specified as both a manually-entered command in your job stream *and* in an action profile or user action definition, then !DB/SMU ignores the manually-entered command.

Specifying a numeric value for the STACK command

STACK can have a value from 1 through 32,768. Use this chart to determine how to specify a value for STACK.

How STACK is Used	How to Specify a Value for STACK
You manually add the STACK command to your generated job stream.	Include the STACK command inline preceding the SCAN, SCANR, or MONITOR command in the form: **STACK *nn*** where *nn* is a value from 1 through 32,768. Do not include the equals sign (=).
You add the STACK command to an action profile you define on the Action Profile *profilename* panel. When included, it must be the *last* command in the inline user action definition.	!DB/SMU displays a pop-up panel on which you can specify a numeric limit for STACK from 1 through 32,768.
You include the STACK command as part of your inline user action definition. When included, it must be the *last* command in the inline user action definition.	Include the STACK command inline in the form: **STACK=*nn*** where *nn* is a value from 1 through 32,768.

What if the STACK limit is not reached

It is possible that the total number of objects queued for a given action is less than the number you have specified for the STACK command. This situation can occur, for example, when nearing the end of processing an object list. In this case, a job is created that processes any spaces remaining unprocessed.

A reminder about the Job Control field on the General Options Panel

To understand how the STACK command affects the creation of jobs or job steps, it is necessary to understand its interaction with the Job Control value that you specify on the General Options panel when creating a utility profile for a DB2 utility. As a reminder, the possible values for the Job Control value are:

1 Causes the DB2 utility to process all of the spaces in the utility's queue in 1 job step

2 Causes the DB2 utility to process each of the spaces in the utility's queue in separate job steps

3 Causes the DB2 utility to process each space in the queue as a separate job

Examples of the Interaction between the STACK Command and Job Control Field

Overview

The easiest way to understand the effect of the interaction of the value you specify in the Job Control field and the value you specify for the STACK command is to review several examples.

Example 1

Assume the following:

- Create a user action definition for REORG. The exception condition that triggers the execution of the user action definition REORG is that an object has more than 10 pointer records.

```
POINTERRECORDS>10 REORG
REORG
   NOEXEC
   UTIL COPY
   EXEC
   UTIL REORG MOD
STACK=3
```

- Six objects, A, B, C, D, E, and F, have more than 10 pointer records.
- STACK has been set to 3 (see the example above).

Refer to the chart to determine the effect of the value for the Job Control field:

Value for Job Control Field	What !DB/SMU Creates
Job Control = 1, multiple objects per job step	Two jobs are created, each with two job steps (1 job step for image copy and 1 job step for REORG). Since STACK=3, the number of "multiple objects" that can be handled in a job step is 3. • Job 1 performs an image copy for objects A, B, and C in the first job step and REORGs A, B, and C in the second job step. • Job 2 performs an image copy for objects D, E, and F in the first job step and REORGs D, E and F in the second job step.

Example 1 (continued)

Value for Job Control Field	What !DB/SMU Creates
Job Control = 2, single object per job step	Two jobs are created, each with six job steps (3 job steps for image copy and 3 job steps for REORG). Because STACK=3, three objects from the queue are passed to a job, but each object is handled as a separate job step. Two jobs are created. • Job 1 performs an image copy for object A, an image copy for object B, and an image copy for object C in the first three job steps and REORGs A, REORGs B, and REORGs C in the fourth through sixth job steps. • Job 2 performs an image copy for object D, an image copy for object E, and an image copy for object F in the first three job steps and REORGs D, REORGs E, and REORGs F in the fourth through sixth job steps.
Job Control = 3, single object per job	In this instance, STACK is irrelevant. Twelve jobs are created, each one having a single job step that either image copies an object or REORGs the object. That is, job 1 image copies object A, job 2 REORGs object A, job 3 image copies object B, and so on.

Example 2

Now, assume the same JCL, but let STACK default to 1. Refer to the chart to determine the effect of the value for the Job Control field:

Value for Job Control field	What !DB/SMU Creates
Job Control = 1, multiple objects per job step	Six jobs are created, one per object. STACK=1 causes the six objects to be passed one at a time. Each job contains two job steps, one to image copy the object and the second to REORG the object. Thus job 1 image copies and REORGs object A, job 2 image copies and REORGs object B, and so on.
Job Control = 2, single object per job step	Processing in this case is identical to the case where Job Control = 1. Six jobs are created, one per object. STACK=1 causes the six objects to be passed one at a time. Each job contains two job steps, one to image copy the object and the second to REORG the object. Thus job 1 image copies and REORGs object A, job 2 image copies and REORGs object B, and so on.
Job Control = 3, single object per job	Twelve jobs are created, each having a single object and a single action. Job 1 image copies object A and job 2 REORGs object A; job 3 image copies object B and job 4 REORGs object B; and so on.

Examples of User Action Definitions

Overview

This unit contains examples of user action definitions. A user action definition is placed in line in the batch JCL following the EXCPTCTL DD statement.

Example of a user action definition for multiple actions

The following example contains comments and statements for a sample user action definition. The example initiates series of actions for the user-defined REORG action if the number of pointer records exceeds 10. (See "Introduction to Handling Exception Conditions in Batch Jobs" on page 509 for details about exception processing.)

```
POINTERRECORDS>10  REORG
***
REORG
* SET UTILITY OPTIONS
  UTIL OPTS OPT1
* SUBMIT NO JOBS
  NOEXEC
* BEGIN A NEW JOBSTREAM
  UTIL COPY
  UTIL REORG MOD
  UTIL COPY MOD
  UTIL MODIFY MOD
* SUBMIT JOB NEXT TIME
  EXEC
  UTIL RUNSTATS MOD
* SAVE THIS JOBSTREAM
  PDSCOPY TO=user.dataset(@REORGTS) MOD
* SEPARATE JOB FOR REBIND (NO MOD)
  UTIL REBIND PLANS
* SAVE THIS JOBSTREAM
  PDSCOPY TO=user.dataset(@RBINDTS) MOD
* ISSUE STACK COMMAND
STACK=05
* END OF JOB
END
```

What if no user action definition is found in the inline JCL?

If you do not include the statements comprising a user action definition inline in your !DB/SMU batch JCL, then !DB/SMU will invoke the DB2 utility having the same name as your action definition. In the above example, if the statements from the REORG statement to the END statement did *not* appear, !DB/SMU would invoke the DB2 REORG utility.

Example EXCPTCTL processing for rebinding plans and packages

The following example contains statements for a user action definition that includes the REBIND action for EXCPTCTL processing. The **REBIND PLANS** action applies to both plans and packages.

```
EXCPTCTL
    PAGES > 0 REORG
    REORG
    NOEXEC
    UTIL OPTS UUT10B
    UTIL REORG
    UTIL REBIND PLANS MOD
    PDSCOPY TO=your.dataset(member)
```

Chapter 31.
Using the Batch Reports for !DB/SMU

Introduction

The batch jobs for !DB/SMU print a variety of reports. This chapter covers these reports and the information they give you.

Chapter Contents

Introduction to the Batch Reports

Overview

!DB/SMU generates many reports for batch jobs. This unit describes those reports.

Purpose of the batch reports

Running !DB/SMU in batch provides you with a means to manage a large number of spaces. The reports that !DB/SMU produces perform the following tasks:

- provide a summary about each table space and index space processed by a !DB/SMU batch job
- identify data integrity errors
- identify DB2 spaces that need attention
- identify DB2 spaces that exceed performance and operating criteria
- provide a summary about each processed table space and index space
- provide detailed statistics about table spaces and index spaces for performance analysis

How to generate various batch reports

The following chart describes each of the batch reports you can produce using !DB/SMU. In order to produce the report, you must include the DD statement shown in the "Required DD Statement" column in the batch JCL you use to execute the !DB/SMU batch utility. !DB/SMU writes the necessary information to the dataset represented by the required DD statement.

Name of the Batch Report	Description	Required DD Statement
Copy Factor Report	Lists the datasets according to their copy factor	MNTFILE DD statement generated for a monitor run profile
DASD Extents Report	Lists the datasets by the number of DASD extents the have	MNTFILE DD statement generated for a monitor run profile
Dataset Size Report	Lists the datasets according to size	MNTFILE DD statement generated for a monitor run profile

How to generate various batch reports (continued)

Name of the Batch Report	Description	Required DD Statement
Extent Map Report	Contains information on the DASD extents for any data-sets that were opened	XTNTMAP DD statement generated for any run profile or CLIST that runs the batch utility
Page Errors and Exceptional Conditions Report	Lists the data integrity errors and any exception conditions and the DB2 utilities or actions they triggered	ERRORS DD statement generated for: • Any job using defaults for exception conditions • A monitor run profile
Reorg Factor Report	Lists the datasets according to their reorg factor	MNTFILE DD statement generated for a monitor run profile
Statistics Reports	Contains detailed statistics for specified table spaces and index spaces. (Equivalent to the online scan reports)	STATREPT DD statement generated for a scan run profile
Summary Reports	Contains a 1-line summary of each table space and index space that was processed (!DB/SMU produces two versions of this report, one for full scans and one for space map scans.)	KTSPRINT DD statement generated for a scan run profile or space map run profile

Examples of !DB/SMU batch reports

This chapter contains an example of each batch report. Refer to the following chart to find the example you want.

For an Example of the. . .	Refer To. . . .
Copy Factor Report	"Examples of the Reports for a Monitor Run Profile" on page 598
DASD Extents Report	"Examples of the Reports for a Monitor Run Profile" on page 598
dataset Size Report	"Examples of the Reports for a Monitor Run Profile" on page 598
Extent Map Report	"Example of an Extent Map Report" on page 595
Page Errors and Exceptional Conditions Report	"Example of a Page Errors and Exceptional Conditions Report" on page 601
Reorg Factor Report	"Examples of the Reports for a Monitor Run Profile" on page 598
Statistics Reports	"Examples of the Statistics Reports" on page 605
Summary Reports	"Example of a Summary Report for a Full Batch Scan" on page 626 and "Example of a Summary Report for a Batch Space Map Analysis" on page 628

Example of an Extent Map Report

Overview

Batch jobs for a scan or space map run profile generate an extent map report. This unit contains an example of this report.

Purpose

An extent map report displays DASD extent data for each DB2 dataset that was processed in the batch job.

Elements of an extent map report

This figure shows an example of an extent map report.

```
MM/DD/YY     15:33:58     ----   D B / S M U  V 500  DB2=D31A ---

                                 STARTING      ENDING
DSNAME    EXTENT #   VOLSER      CYL HD      CYL HD     TRKS    CYLS
 1          2          3           4           5          6       7
          --------   ------   ---------   ---------   ------  ------
DSNH.DSNDBD.DSNDB06.SYSCOPY.I0001.A001  1
              1     270025       86  5       86  8        4


DSNH.DSNDBD.DSNDB06.DSNUCH01.I0001.A001
              1     270025       12 13       12 14        2


DSNH.DSNDBD.DSNDB06.SYSDBASE.I0001.A001
              1     270025      189  0      197 14      135       9
              2     270025      107  0      112 14       90       6

 8 TTL        2                                         225      15

DSNH.DSNDBD.DSNDB06.DSNATX01.I0001.A001
              1     270025      134  5      134 11        7
              2     270025      203  9      203 13        5

   TTL        2                                          12

DSNH.DSNDBD.DSNDB06.SYSPLAN.I0001.A001
              1     270025      199  0      200 14       30       2
              2     270025      402  0      403 14       30       2
              3     270025      404  0      405 14       30       2
              4     270025      422  0      423 14       30       2
              5     270025      162  0      163 14       30       2

   TTL        5                                         150      10
```

Elements of an extent map report (continued)

1 Names of the DB2 datasets that were processed in the batch job. All the names have the following format:

```
vcatname.DSNDBD.dbname.spacename.I0001.A0nn
```

Where:

vcatname Name of the VSAM catalog that is assigned to this space

dbname Name of the DB2 database

spacename Name of the table space or index space

nn Partition number, or 01 if not partitioned

2 Number of the DASD extent

3 Serial number of the DASD volume for the extent

4 DASD cylinder and head number of the first track of the extent. The values are decimal. The first cylinder on each volume is cylinder 0; the first head on each cylinder is head 0.

5 DASD cylinder and head number of the final track of the extent

6 Number of tracks in this extent

7 If the space is allocated in cylinder increments, the number of cylinders in this extent

8 When a dataset spans multiple extents, the totals for number of extents, tracks, and cylinders

Examples of the Reports for a Monitor Run Profile

Overview

You can request four optional reports when you define a monitor run profile. This unit contains examples of these reports.

The optional monitor run profile reports

When you create a monitor run profile, you can specify that any or all of these optional reports be produced:

- Dataset size report
- Reorg factor report
- Copy factor report
- DASD extents report

Each report lists the datasets in either ascending or descending order according to size, reorg or copy factor, or number of extents.

Elements of the reports

All of the reports contain the same elements:

1 DB2 database name

2 Table space name

3 Partition number, if applicable

4 Index space name

5 Date on which extract was run

6 DB2 catalog name

7 Name of the creator and table for the space (table space or index space)

8 Size of the dataset

9 Number of DASD extents occupied by the space

10 Copy factor for the space

11 Reorg factor for the space

This is an example dataset size report in ascending order.

```
MM/DD/YY      15:33:58        ----  D B / S M U  V 500  DB2=D31A ----                    PAGE 7

       --------------------    REPORT IN ASCENDING SIZE ORDER            --------------------

                                EXTRACT                                                DATASET   DASD      COPY  REORG
DBDNAME  TBLSPACE PT IDXSPACE PT DATE     CATALOG  CREATOR-TABLE                          SIZE  EXTENTS  FACTOR  FACTOR
   1        2     3     4     3    5         6        7                                     8       9       10      11
-------- -------- -- -------- -- -------- -------- ------------------------- ---------   ------- ------- -------
DBKENT   PLANRTAB                04/20/99 TDDB231A TSL251                               12 P     1      100.0    16.5
DSNDB06  SYSDBASE    DSNDLX01    07/22/99 TDDB231A SYSIBM.SYSRELS                       12 P     1       58.5
         SYSDBAUT    DSNADH01    07/22/99 TDDB231A SYSIBM.SYSDBAUTH                     12 P     1       41.5
                     DSNADX01    07/22/99 TDDB231A SYSIBM.SYSDBAUTH                     12 P     1       25.0
                     DSNDDH01    07/22/99 TDDB231A SYSIBM.SYSDATABASE                   12 P     1       25.0
         SYSGROUP    DSNSSH01    07/22/99 TDDB231A SYSIBM.SYSSTOGROUP                   12 P     1       25.0
         SYSPKAGE    DSNKLX01    07/22/99 TDDB231A SYSIBM.SYSPACKLIST                   12 P     1       25.0
                     DSNKPX01    07/22/99 TDDB231A SYSIBM.SYSPLSYSTEM                   12 P     1       41.5
         SYSSTR      DSNSSX01    07/22/99 TDDB231A SYSIBM.SYSSTRINGS                    12 P     1       25.0
KCS01DB  KTSSQLDD                07/20/99 TDDB231A CANDLE1                              12 P     1      100.0    16.5
         KTSSQLDE                07/20/99 TDDB231A CANDLE1                              12 P     1      100.0    16.5
KTCDBD   STATDBAS    ALTC15N#    08/19/99 TDDB231A CDB.ALTERCONTROL                     12 P     1       25.0
                     ALTM1V6K    05/26/99 TDDB231A CDB.ALTERMNQUEUE                     12 P     1       25.0
KTSDBD31             ALTCTRLX    05/26/99 TDDB231A CANDLE1.ALESSAGES                    12 P     1       25.0
                     ALTENQUX    05/26/99 TDDB231A CANDLE1.ALTERCONTROL                 12 P     1       25.0
                     ALTMESGX    05/26/99 TDDB231A CANDLE1.ALTERENQUEUE                 12 P     1       25.0
TTTDEXP  ACT                     08/10/99 TDDB231A CANDLE2.TERMESSAGES                  12 P     1       25.0
KTCDBD   STATDBAS    ALTCTRLX    05/26/99 TDDB231A CANDLE4.ALT                          12 P     1      100.0    75.0
                     ALTENQUX    05/26/99 TDDB231A CANDLE4.ALTERCONTROL                 20 P     1       15.0
                     ALTMESGX    05/26/99 TDDB231A CANDLE4.ALTERENQUEUE                 20 P     1       15.0
KTSDBD               ALTCTRLX    05/26/99 TDDB231A CANDLE3.ALTERMESSAGES                20 P     1       15.0
                     ALTENQUX    05/26/99 TDDB231A CANDLE3.ALTERCONTROL                 20 P     1       15.0
                     ALTMESGX    05/26/99 TDDB231A HOUSTON.ALTERENQUEUE                 20 P     1       15.0
CHGDBD               ALTENQUX    05/26/99 TDDB231A HOUSTON.ALTERMESSAGES                20 P     1       15.0
DSNDB06  SYSDBASE    DSNDYX01    07/22/99 TDDB231A SYSIBM.SYSERENQUEUE                  24 P     1       12.5
         SYSGPAUT    DSNAGH01    07/22/99 TDDB231A SYSIBM.SYSSYNONYMS                   24 P     1       54.0
```

This is an example reorg factor report in descending order.

```
MM/DD/YY    15:33:58       ----   D B / S M U  V 500  DB2=D31A ----               PAGE 16

       --------------------   REPORT IN DESCENDING REORG FACTOR ORDER       --------------------  PAGE 16

                               EXTRACT                                          DATASET   DASD     COPY   REORG
DBDNAME TBLSPACE PT IDXSPACE PT DATE     CATALOG  CREATOR-TABLE                   SIZE  EXTENTS  FACTOR  FACTOR
   1       2     3     4     3    5         6        7                              8       9       10      11
------- -------- -- -------- -- -------- -------- ------------------------ -------   ------- ------- -------
        SYSPLAN                07/22/99 TDDB231A SYSIBM                      3,240 P     2      94.5    94.0
DSNDB01 SPT01                  08/29/99 TDDB231A SYSIBM                      1,980 P     3      67.5    75.0
TTTDEXP ACT                    08/10/99 TDDB231A CANDLE                         12 P     1      00.0    75.0
DSNDB01 SCT02                  04/20/99 TDDB231A SYSIBM                      4,500 P     4      66.5    64.0
DSNDB06 SYSPKAGE               07/22/99 TDDB231A SYSIBM                        900 P     1      96.0    60.0
        SYSDBASE               05/23/99 TDDB231A SYSIBM                      5,700 P     5      47.0    47.0
        SYSSTR                 07/22/99 TDDB231A SYSIBM                         24 P     1        .0    46.0
        SYSVIEWS               07/22/99 TDDB231A SYSIBM                      2,520 P     2      38.5    37.0
DSNDB01 DBD01                  04/20/99 TDDB231A SYSIBM                        900 P     2      66.5    26.0
DBKENT  PLANRTAB               04/20/99 TDDB231A CANDLE                         12 P     1      00.0    16.5
DSNDB06 SYSDBAUT               07/22/99 TDDB231A SYSIBM                         36 P     1      33.5    16.5
        SYSGROUP               07/22/99 TDDB231A SYSIBM                         24 P     1      50.0    16.5
KCS01DB KTSSQLDD               07/20/99 TDDB231A CANDLE                         12 P     1      00.0    16.5
        KTSSQLDE               07/20/99 TDDB231A CANDLE                        300 P     1      23.5    16.0
DSNDB01 SYSLGRNG               04/20/99 TDDB231A SYSIBM                         48 P     1      25.0     8.5
DSNDB06 SYSGPAUT               07/22/99 TDDB231A SYSIBM                         60 P     1      20.0     8.5
        SYSUSER                07/22/99 TDDB231A SYSIBM                         24 P     1      50.0     8.5
KCS01DB KCS01SUP  1            07/20/99 TDDB231A CANDLE                         24 P     1      50.0     8.5
                  3            07/20/99 TDDB231A CANDLE                         24 P     1      50.0     8.5
                  7            07/20/99 TDDB231A CANDLE                        792 P     1       7.5     6.0
        KTSEDSCG               07/20/99 TDDB231A CANDLE                        132 P     1      18.0     5.5
DSNDB06 SYSCOPY                07/22/99 TDDB231A SYSIBM                         96 P     1        .0     2.0
KTS01DB KTS01SUP  1            05/27/99 TDDB231A CANDLE                        792 P     1        .0      .5
KCS01DB KTSSQLDY               07/20/99 TDDB231A CANDLEALTERENQUEUE             24 P     1      12.5      .5
CHGDBD  STATDBAS    ALTENQUX   05/26/99 TDDB231A TDDB43SYSCOPY                  36 P     1      30.5
DSNDB06 SYSCOPY     DSNUCH01   07/22/99 TDDB231A SYSIBMSYSCOPY                  60 P     1       8.5
                    DSNUCX01   07/22/99 TDDB231A SYSIBMSYSONTROL
```

This is an example copy factor report in descending order.

```
 MM/DD/YY      15:33:58         ----   D B / S M U  V 500  DB2=D31A ----               PAGE 16

        --------------------    REPORT IN DESCENDING COPY FACTOR ORDER          --------------------

                                EXTRACT                                      DATASET   DASD     COPY   REORG
DBDNAME TBLSPACE PT IDXSPACE PT DATE     CATALOG  CREATOR-TABLE                 SIZE  EXTENTS FACTOR  FACTOR
  1       2      3    4       3  5        6        7                          8       9      10      11
------- -------- -- -------- -- -------- -------- ------------------------ -------  ------- ------ -------
DSNDB06 SYSPKAGE    DSNKDX02    07/22/99 TDDB231A SYSIBM.SYSPACKDEP            36 P  2       58.5
KCS01DB KTSEDSCG    KTSEDI22    07/20/99 TDDB231A CANDLE1.KTSEDC22             24 P  1       58.5
DSNDB06 SYSPLAN     DSNAPH01    07/22/99 TDDB231A SYSIBM.SYSPLANAUTH          108 P  2       56.5
        SYSDBASE    DSNDYX01    07/22/99 TDDB231A SYSIBM.SYSSYNONYMS           24 P  1       54.0
        SYSGROUP                07/22/99 TDDB231A SYSIBM                       24 P  1       50.0    16.5
        SYSPKAGE    DSNKYX01    07/22/99 TDDB231A SYSIBM.SYSPKSYSTEM           24 P  1       50.0
KCS01DB KCS01SUP  1             07/20/99 TDDB231A CANDLE1                      24 P  1       50.0     8.5
                  3             07/20/99 TDDB231A CANDLE1                      24 P  1       50.0     8.5
                  7             07/20/99 TDDB231A CANDLE1                      24 P  1       50.0     8.5
DSNDB06 SYSDBASE                05/23/99 TDDB231A SYSIBM                    5,700 P  5       47.0    47.0
        SYSDBAUT    DSNADH01    07/22/99 TDDB231A SYSIBM.SYSDBAUTH             12 P  1       41.5
        SYSPKAGE    DSNKPX01    07/22/99 TDDB231A SYSIBM.SYSPLSYSTEM           12 P  1       41.5
        SYSVIEWS                07/22/99 TDDB231A SYSIBM                    2,520 P  2       38.5    37.0
        SYSDBASE    DSNDCX01    07/22/99 TDDB231A SYSIBM.SYSCOLUMNS         2,160 P  1       36.5
        SYSDBASE    DSNDCX01    07/22/99 TDDB231A SYSIBM.SYS                   36 P  1       33.5    16.5
        SYSDBAUT                07/22/99 TDDB231A SYSIBM.COPY                  36 P  1       30.5
        SYSCOPY     DSNUCH01    07/22/99 TDDB231A SYSIBM.SYSVIEWDEP            48 P  1       29.0
        SYSVIEWS    DSNGGX02    07/22/99 TDDB231A SYSIBM.SYSDBAUTH             12 P  1       25.0
        SYSDBAUT    DSNADX01    07/22/99 TDDB231A SYSIBM.SYSDATABASE           12 P  1       25.0
                    DSNDDH01    07/22/99 TDDB231A SYSIBM.SYS                   48 P  1       25.0     8.5
        SYSGPAUT                07/22/99 TDDB231A SYSIBM.RESAUTH               24 P  1       25.0
                    DSNAGH01    07/22/99 TDDB231A SYSIBM.SYSSTOGROUP           12 P  1       25.0
        SYSGROUP    DSNSSH01    07/22/99 TDDB231A SYSIBM.SYSPACKLIST           12 P  1       25.0
        SYSPKAGE    DSNKLX01    07/22/99 TDDB231A SYSIBM.SYSSTRINGS            12 P  1       25.0
        SYSSTR      DSNSSX01    07/22/99 TDDB231A SYSIBM.SYSONTROL             12 P  1       25.0
KTCDBD  STATDBAS    ALTC15N#    08/19/99 TDDB231A CDB.ALTERCNQUEUE             12 P  1       25.0
                    ALTE1AFF    05/26/99 TDDB231A CDB.ALTEREESSAGES            12 P  1       25.0
                    ALTM1V6K    05/26/99 TDDB231A CDB.ALTERMESSAGES            12 P  1       25.0
```

This is an example of a DASD extents report in descending order.

```
 MM/DD/YY      15:33:58         ----   D B / S M U  V 500  DB2=D31A ----               PAGE 16

        --------------------    REPORT IN DESCENDING NUMBER OF EXTENTS ORDER      ----------------------  PAGE 16

                                EXTRACT                                      DATASET    DASD      COPY   REORG
DBDNAME TBLSPACE PT IDXSPACE PT DATE     CATALOG  CREATOR-TABLE                 SIZE   EXTENTS  FACTOR  FACTOR
  1       2      3    4       3  5        6        7                          8        9       10      11
------- -------- -- -------- -- -------- -------- ------------------------ -------   ------- ------- -------
DSNDB06 SYSDBASE                05/23/99 TDDB231A SYSIBM                    5,700 P    5       47.0    47.0
DSNDB01 SCT02                   04/20/99 TDDB231A SYSIBM                    4,500 P    4       66.5    64.0
DSNDB06 SYSPKAGE    DSNKDX01    07/22/99 TDDB231A SYSIBM.SYSPACKDEP           276 P    4       98.0
                    DSNKKX02    07/22/99 TDDB231A SYSIBM.SYSPACKAGE            96 P    4       94.0
                    DSNKSX01    07/22/99 TDDB231A SYSIBM.SYSPACKSTMT          384 P    4       96.0
DSNDB01 SPT01                   08/29/99 TDDB231A SYSIBM                    1,980 P    3       67.5    75.0
DSNDB06 SYSPKAGE    DSNKAX03    07/22/99 TDDB231A SYSIBM.SYSPACKAUTH          120 P    3       73.5
DSNDB01 DBD01                   04/20/99 TDDB231A SYSIBM                      900 P    2       66.5    26.0
DSNDB06 SYSDBASE    DSNDSX01    07/22/99 TDDB231A SYSIBM.SYSTABLESPACE        120 P    2       96.0
                    DSNDTX02    07/22/99 TDDB231A SYSIBM.SYSTABLES             96 P    2       92.5
                    DSNDXX02    07/22/99 TDDB231A SYSIBM.SYSINDEXES            72 P    2       89.0
        SYSPKAGE    DSNKDX02    07/22/99 TDDB231A SYSIBM.SYSPACKDEP            36 P    2       58.5
                    DSNKKX01    07/22/99 TDDB231A SYSIBM.SYSPACKAGE            96 P    2       95.0
                    DSNKLX02    07/22/99 TDDB231A SYSIBM.SYSPACKLIST          120 P    2       96.0
        SYSPLAN                 07/22/99 TDDB231A SYSIBM                    3,240 P    2       94.5    94.0
                    DSNAPH01    07/22/99 TDDB231A SYSIBM.SYSPACKAUTH          108 P    2       56.5
        SYSSTATS    DSNTCX01    07/22/99 TDDB231A SYSIBM.SYSPACKAUTH        1,080 P    2         .5
                    DSNTTX01    07/22/99 TDDB231A SYSIBM.SYSPACKLIST        1,080 P    2         .5
        SYSVIEWS                07/22/99 TDDB231A SYSIBM    PLSYSTEM        2,520 P    2       38.5    37.0
CHGDBD  STATDBAS    ALTENQUX    05/26/99 TDDB231A CANDLE.ALTPKSYSTEM           24 P    1       12.5
DBKENT  PLANRTAB                04/20/99 TDDB231A TSL251    PLANAUTH           12 P    1      100.0    16.5
DSNDB01 SYSLGRNG                04/20/99 TDDB231A SYSIBM    PLANDEP           300 P    1       23.5    16.0
DSNDB06 SYSCOPY                 07/22/99 TDDB231A SYSIBM    PLAN              132 P    1       18.0     5.5
                    DSNUCH01    07/22/99 TDDB231A SYSIBM.SYS                   36 P    1       30.5
                    DSNUCX01    07/22/99 TDDB231A SYSIBM.SYSCOLDIST            60 P    1        8.5
        SAVDBASE    DSNATX01    07/22/99 TDDB231A SYSIBM.SYSCOLDISTSTATS      144 P    1       74.5
                    DSNATX02    07/22/99 TDDB231A SYSIBM.SYSINDEXSTATS        600 P    1       85.0
```

Example of a Page Errors and Exceptional Conditions Report

Overview

A batch job for a scan run profile generates a page errors and exceptional conditions report. This unit contains an example of this report.

Purpose of the report

The page errors and exceptional conditions report lists any data integrity errors that !DB/SMU detected during a job and any exception conditions and actions that they triggered. !DB/SMU saves this information in the system PDS. This enables you to use the online repair facility to correct errors.

The report identifies datasets that exceed your standards. Use this information to focus your attention on *only* those datasets. Use the information about reported exception conditions to reduce the workload associated with database administration and DASD management.

Elements of a page errors and exceptional conditions report

The example shows a page errors and exceptional conditions report.

```
                            ----   D B / S M U   ----  DB2=DSN2 ----              PAGE 1

PAGE ERRORS AND EXCEPTIONAL CONDITIONS THAT REQUIRE ATTENTION
------------------------------------------------------------------------------------------
1 DBD=QGHIJKXX, TABLESPACE=SPRTAE87  EXCEPTIONS
                              2                  3                         4
              VALUE =         2,590, CONDITION = UNFORMATTEDPAGES>=1000
              VALUE =         4,608, CONDITION = UPDATEDPAGES>=1000       INCRCOP
              VALUE =         36.0%, CONDITION = UNFORMATTEDSPACEPCT>=30
              VALUE =        100.0%, CONDITION = PCTPAGESUPDATED>=30      REORG
              VALUE =        15,090, CONDITION = ROWSDATA>=1000
              VALUE =         17.4%, CONDITION = OVERHEADPCT>=10
              VALUE =        10,362, CONDITION = FREESPACE>=1000
              VALUE =        10,360, CONDITION = UNFORMATTEDSPACE>=1000
              VALUE =        28,800, CONDITION = TOTALSPACE>=1000
              VALUE =          64.0, CONDITION = COPYFACTOR>=40           FULLCOP
              VALUE =          64.0, CONDITION = COPYFACTOR>=10           INCRCOP

1 DBD=QGHIJKXX, INDEXSPACE=IRTAEM87  EXCEPTIONS
              VALUE =       396,225, CONDITION = RECORDSINDEXED>=1000
              VALUE =         2,096, CONDITION = USEDSUBPAGES>=1000
              VALUE =         3,716, CONDITION = TOTALSUBPAGES>=1000
              VALUE =         1,620, CONDITION = EMPTYSUBPAGES>=1000      REORG
              VALUE =         1,000, CONDITION = TOTALINDEXPAGES>=1000
              VALUE =         54.9%, CONDITION = PCTLEAFFREESPACE>=25     REORG
              VALUE =       2,358.5, CONDITION = AVGRECORDSPERKEY>=1000

1 DBD=QGHIJKXX, INDEXSPACE=IRTAES87  EXCEPTIONS
              VALUE =       396,225, CONDITION = RECORDSINDEXED>=1000
              VALUE =       204,333, CONDITION = DISTINCTKEYS>=1000
              VALUE =         2,093, CONDITION = LEAFPAGES>=1000
              VALUE =         2,106, CONDITION = USEDINDEXPAGES>=1000
              VALUE =         8,338, CONDITION = USEDSUBPAGES>=1000
              VALUE =         8,372, CONDITION = TOTALSUBPAGES>=1000
              VALUE =         2,250, CONDITION = TOTALINDEXPAGES>=1000
              VALUE =         44.5%, CONDITION = PCTLEAFFREESPACE>=25     REORG
```

Elements of a page errors and exceptional conditions report (continued)

```
PAGE ERRORS AND EXCEPTIONAL CONDITIONS THAT REQUIRE ATTENTION                      PAGE 2
-----------------------------------------------------------------------------------------
1 DBD=QGHIJKXX, TABLESPACE=SPRTAE88  EXCEPTIONS
                               2                   3                         4
              VALUE =         3,115, CONDITION = UNFORMATTEDPAGES>1000
              VALUE =         4,083, CONDITION = UPDATEDPAGES>=1000       INCRCOP
              VALUE =         4,083, CONDITION = PAGESOCCUPIED>=1000
              VALUE =         43.3%, CONDITION = FREESPACEPCT>=40         REORG
              VALUE =         43.3%, CONDITION = UNFORMATTEDSPACEPCT>=30
              VALUE =        100.0%, CONDITION = PCTPAGESUPDATED>=30      REORG
              VALUE =        13,332, CONDITION = ROWSDATA>=1000
              VALUE =         2,902, CONDITION = OVERHEAD>=1000
              VALUE =        12,461, CONDITION = FREESPACE>=1000
              VALUE =        12,460, CONDITION = UNFORMATTEDSPACE>=1000
              VALUE =        28,800, CONDITION = TOTALSPACE>=1000
              VALUE =          56.8, CONDITION = COPYFACTOR>=40           FULLCOP
              VALUE =          56.8, CONDITION = COPYFACTOR>=10           INCRCOP

1 DBD=QGHIJKXX, INDEXSPACE=IRTAEM88  EXCEPTIONS
              VALUE =       359,267, CONDITION = RECORDSINDEXED>=1000
              VALUE =         1,961, CONDITION = USEDSUBPAGES>=1000
              VALUE =         3,364, CONDITION = TOTALSUBPAGES>=1000
              VALUE =         1,403, CONDITION = EMPTYSUBPAGES>=1000      REORG
              VALUE =         54.8%, CONDITION = PCTLEAFFREESPACE>=25     REORG
              VALUE =       8,355.0, CONDITION = AVGDISTINPAGES>=171

1 DBD=QGHIJKXX, INDEXSPACE=IRTAES88  EXCEPTIONS
              VALUE =       359,267, CONDITION = RECORDSINDEXED>=1000
              VALUE =       208,835, CONDITION = DISTINCTKEYS>=1000
              VALUE =         2,043, CONDITION = LEAFPAGES>=1000
              VALUE =         2,056, CONDITION = USEDINDEXPAGES>=1000
              VALUE =         8,139, CONDITION = USEDSUBPAGES>=1000
              VALUE =         8,172, CONDITION = TOTALSUBPAGES>=1000
              VALUE =         2,100, CONDITION = TOTALINDEXPAGES>=1000
              VALUE =         44.3%, CONDITION = PCTLEAFFREESPACE>=25     REORG

1 DBD=QGHIJKXX, INDEXSPACE=IRTASR88  EXCEPTIONS
              VALUE =       359,267, CONDITION = RECORDSINDEXED>=1000
              VALUE =        15,317, CONDITION = DISTINCTKEYS>=1000
              VALUE =         2,646, CONDITION = USEDSUBPAGES>=1000
              VALUE =         2,768, CONDITION = TOTALSUBPAGES>=1000
              VALUE =         35.1%, CONDITION = PCTLEAFFREESPACE>=25     REORG
              VALUE =          31.1, CONDITION = AVGDISTINPAGES>=30       REORG
```

1 Identification of the space. The report contains an entry for each table space and index space that met or exceeded the exception conditions where:

DBD=	Name of the DB2 database
Table space=	Name of the table space
Indexspace=	Name of the index space

2 The value you specified when defining your exception conditions.

3 The control statement that defines the condition, its trigger value

4 The action !DB/SMU took because an exception condition was met

Considerations for interpreting a page errors and exceptional conditions report

When fractional values are reported, they are multiplied by 10. For example, a value of 171 for AVGDISTINPAGES in the report actually equals 17.1. You may think that a trigger limit was exceeded without initiating the required action. This is not true case. If no action is reported, the actual value is within limits.

Exception Condition	Report Example
The exception profile initiates the REORG utility if the average distance in between pages is greater than or equal to 30 (AVGDISTINPAGES >=30 REORG).	The report shows a value of 215 and indicates that no action was taken. The actual value, however, is 21.5, which is less than an average distance of 30 between pages.

Examples of the Statistics Reports

Overview

A batch job for a scan run profile generates a set of comprehensive statistics reports. This unit provides information and examples for these reports.

Content of the reports

These reports contain detail information about table spaces and index spaces. They are similar to those for online scans. The set contains those reports that are appropriate for the specified spaces. For example, if there was insufficient data to generate a graph, then the graph would be excluded from the reports. Likewise, reports on data compression are excluded if a table space is not compressed.

Statistics reports for table spaces

This chart summarizes the types of statistics reports that batch scans produce for table spaces.

Name of Report	Description
Compressed Rows Summary	Contains information on compressed rows and pages for each table
Compressed Row Usage	Contains statitstics for compression and page savings
DASD SPACE Analysis	Describes DASD usage
Dictionary	Contains information on overhead for the DB2 dictionary
Freespace Analysis	Reports on available free space
Pages Analysis	Contains comprehensive statistics for the pages in the table space
Pointer Analysis	Reports on pointer and overflow records
Tables Analysis	Contains information on compression, rows, and pages for each table

Graphs for table space statistics

This chart summarizes the types of graphs that batch scans produce for table spaces.

Type of Graph	Description
Cumulative distribution of free space	Shows the total percentage of free space at any specific point in the table space
Cumulative distribution of updated pages	Shows the total percentage of updated pages at any specific point in the table space
Free space in pages	Shows the location of pages that have room for 1 or more new rows
Free space in the table	Shows the location of free space in the table
Updated pages	Shows the location of updated pages

Statistics reports for index spaces

This chart summarizes the types of statistics reports that batch scans produce for index spaces.

Name of Report	Description
Index Keys Summary	Contains statistics that summarize the characteristics of the index
Index Level Summary	Contains statistics for pages, keys, and free space
Pages Summary	Shows the allocation of pages within the index
Subpages Summary	Shows the use of subpages

Graphs for index spaces

This chart summarizes the types of graphs that batch scans produce for index spaces.

Type of Graph	Description
Index free space	Shows the location of free space in the index
Index usage	Shows the location of used index pages
Pages with room for keys	Shows the location of index pages that have room for keys
Records per key	Shows the location of records per key
Ten most frequent keys	Shows the 10 keys that have the most records

Example of the statistics reports for a full batch scan

The following pages contain example statistics reports for table spaces and index spaces.

```
MM/DD/YY     1 5:33:58        ----   D B / S M U  V XXX  DB2=D31A ----              PAGE 16

                                          D B / S M U    STATISTICS  REPORTS         DBD = KCS01DB     TABLESPACE = KTS12SCG

                                      ****************************************************
                                      *                                                  *
                                      *       DBD = KCS01DBX    TABLESPACE = KTS12SCG    *
                                      *                                                  *
                                      ****************************************************

PAGES ANALYSIS
                             NUMBER        PCT OF
                           OF PAGES       ALL PGS
                          ---------       -------                                                       4   PAGES WITH ERRORS 10
          PAGES WITH ROWS       186        23.48%  1
        DB2 CONTROL PAGES         2         0.25%  2
     DB2 DICTIONARY PAGES        16         2.02%  3
              EMPTY PAGES        50         6.31%  4
        UNFORMATTED PAGES       543        68.56%  5

                    TOTAL       792       100.00%                                          PAGES NEEDED AFTER REORG  237      11


            UPDATED PAGES         2         0.25%  6
           UPDATED TRACKS         1                7                                         COPY FACTOR =   1.5            12
      UPDATED EMPTY PAGES         0                8
          POINTER RECORDS         0                9

------------------------------------------------------------------------------------------------------------------------------------------------

FREESPACE ANALYSIS

                             NUMBER        PCT OF         ROW        PAGE
                           OF PAGES       ALL PGS        SIZE        SIZE
                          ---------       -------        ----        ----
      PAGES WITH:
                  NO ROOM        26 1      3.28%                   4,096              FREESPACE SPECIFIED    50%  5
       ROOM FOR SHORT ROW         0 2        0%           74                           FREEPAGE SPECIFIED     4   6
     ROOM FOR AVERAGE ROW         0 3        0%           74
        ROOM FOR LONG ROW       766 4     96.72%          74                            REORG FACTOR =       3.3  7
```

Page 1 of the STATREP reports contains samples of the Pages Analysis and Free Space Analysis reports.

Pages Analysis Report

The Pages Analysis report summarizes the status of all the pages in the table space. This report contains the following:

1 Number of pages that contain data

2 Number of DB2 control pages (Includes header page and space map pages)

3 Number of dictionary pages ESA compressed space (Printed if the space is compressed)

4 Number of empty, formatted pages that no longer hold data or that never held data

5 Number of empty, unformatted pages never used by DB2

6 Number of updated pages

7 Number of updated tracks

8 Number of pages whose rows have been deleted since the last image copy (Indicates the degree of delete activity)

9 Number of pointer records

10 Number of pages with errors (Only printed if the scan finds errors)

11 Number of pages estimated for the table space if REORG is run immediately after the scan

12 Percentage of updated tracks (Indicates the need to make an image copy.)

Analysis

The low Copy Factor (1.5) shows that only 1 track was updated; an image copy is unecessary.

Pointer Analysis Report

The Pointer Analysis report summarizes the number of pointer records and overflow records in a table space. This report was not printed because there were no pointer records in the table space.

Free Space Analysis Report

The Free Space Analysis report gives information on the amount of available free space. This report contains the following information:

1 Number of pages that have no room

2 Number of pages have room for only the shortest row size of all the tables in the table space (If this value is 0, the table space contains only 1 table.)

3 Number of pages that have room for the average row size of all the table spaces (If this value is 0, the table space contains only 1 table.)

4 Number of pages that have room for the longest row size of all the tables in the table space (If the table space contains multiple tables, !DB/SMU computes the size of the average, longest, and shortest row sizes in the table space.)

5 Free space value in effect during the scan

6 Free page value in effect during the scan

7 Percentage of pages with no room (Indicates the fullness of the table space and the need to reorganize it)

N/A The status of ESA compression is printed if the space is compressed.

Analysis

The sample shows the following:

- The table space contains only 1 table because all the rows are the same size (74).
- A low reorg factor (3.3) means that 3.3% of the pages are full and cannot take an insert.
- The pages with no room (26) contain less than 74 bytes of free space.

Reorganization is probably not required as indicated by the low reorg factor and high percentage of pages with room.

MM/DD/YY 15:33:58 ---- D B / S M U V XXX DB2=D31A ---- PAGE 2

PAGE 2

D B / S M U STATISTICS REPORTS DBD = KCS01DB TABLESPACE = KTS12SCG

DASD SPACE ANALYSIS

	NUMBER OF PAGES	AMOUNT OF SPACE	% USED SPACE	% TOTAL SPACE		
ROWS DATA	178	266K	26.50%	8.40%	1	
DB2 DICITIONARY PAGES	24	64K	6.37%	2.02%	2	
WASTED DICTIONARY	8	32K	3.19%	1.01%	3	
PAGES WITH OVERHEAD	222	91K	9.06%	2.87%	4	
PAGES W WASTED SPACE	124	0K	0.05%	0.02%	5	
PAGES WITH FREESPACE	225	550K	54.82%	17.37%	6	
FORMATTED EMPTY	42	167K		5.27%	7	
UNFORMATTED EMPTY	5	20K		0.63%	8	
ALLOCATED / NOT USED	541	2,164K	215.54%	68.31%	9	PCT FREESPACE = 85.7 11
TOTAL DATASET SPACE		3,168K 10				

TABLES ANALYSIS

CREATOR.TABLE	1 ROWS IN TABLE	2 PCT OF TOTAL	3 OVERFLOW ROWS	4 COMPRESS ROWS	5 TOTAL ROW SPACE	6 PCT OF TOTAL	7 PAGES OCCUPIED	8 AVG ROWS PER PAGE	9 AVG LEN PER ROW	10 COMPTD ROW SIZ
CANDLE1.KTS12C22	3,384	32.76%		3,384	26K	9.64%	28	120.86	7.88	66
CANDLE1.KTS12C33	3,384	32.76%		3,384	26K	9.64%	28	120.86	7.88	66
CANDLE1.KTS12C44	3,345	32.38%		0	216K	79.81%	125	26.76	66.00	66
DROPPED TBLS ROWS 11	0									
** TOTALS ** 12	10,329	100.00%		6,972	270K	100.00%	186	55.53	26.78	

Page 2 of the sample report contains the DASD Space Analysis and Tables Analysis reports.

DASD Space Analysis Report

The DASD Space Analysis report shows how the table space uses DASD. This report contains the following information:

1. Total number of pages having rows with data, including pages with dropped rows, and excluding header, space map, and dictionary pages
2. Total number of pages that contain the dictionary (This line appears only for compressed table spaces.)
3. Number of wasted dictionary pages—the number of pages in the segment(s) allocated to the dictionary that do not contain dictionary pages (This line appears only for compressed and segmented table spaces.)
4. Total number of pages in the table space that contain overhead.
5. Total number of pages in the table space that contain wasted space. A page is considered to have wasted space when the amount of free space in the page exceeds the average row size multiplied by the number of open rows.
6. Total number of pages in the table space that contain free space
7. Total number of pages in the table space that are formatted but empty. These pages are present when a page once held data but all of the rows have been deleted, or as the result of a FREEPAGE specification during table space creation.
8. Total number of pages in the table space that are allocated to a table but are unformatted
9. Total number of pages in the table space that are allocated to the table space but not yet processed by DB. This value is *not* included in the pages with free space value.
10. Size of the data set in K-bytes
11. Percentage of free space in both unused and unformatted pages

Analysis

The sample shows the following:

- The moderate overhead percentage (9.06%) indicates rather average rows.
- There is potential for reclaiming DASD as indicated by the large percentage of Allocated/Not Used.

Tables Analysis Report

The Table Space Analysis report shows how each table uses the space. This report contains the following information:

1. Number of rows in each table
2. Percentage of the total number of rows used by each table
3. Number of overflow rows in each table
4. Number of compressed rows in each table
5. Amount of DASD space in K-bytes used by each table
6. Percentage of DASD space used by each table
7. Number of pages in each table space that contain rows of data
8. Number of rows in each table divided by the number of occupied pages and the average number of rows per page for all tables
9. Total length of all rows in the table divided by the number of rows (Useful for estimating space and analyzing compressed spaces)
10. Sum of the defined lengths of all the columns in the table. (VARCHAR columns are counted full length, including the 2-byte length prefix.)
11. Number of rows from dropped tables in the table space. (Dropped tables' rows continue to occupy space until REORG is run)
12. Summary totals for the table space

Analysis

The sample shows the following:

- There are no dropped tables' rows.
- The average number of rows per page (55.53) is the total number of rows (10,329) divided by the total number of pages that contain rows (186).
- The average row length (26.78) is the is the sum of the different row lengths divided by the total number of rows.
- Consider running REORG to rebuild the dictionary so that the 3rd table benefits from compression.

```
MM/DD/YY   15:33:58        ----   D B / S M U  V XXX  DB2=D31A ----                          PAGE 16

                                        D B / S M U    STATISTICS  REPORTS     DBD = KCS01DB      TABLESPACE = KTS12SCG

DICTIONARY REPORT
      TOTAL DICTIONARY PAGES:                    16  1
      MAX ENTRIES PER PAGE:                     508  2
      DICTIONARY PAGE SIZE:                       4  3
      TOTAL 4K MEMORY PAGES REQUIRED:            16  4



-------------------------------------------------------------------------------------------------------------------------------------------------------

COMPRESS ROWS SUM


                                    1          2          3          4
                                  ROWS IN  COMPRESSED  CPR ROW    PAGES
CREATOR.TABLE                       TABLE     ROWS       PCT      OCCUPIED
---------------------------------  --------- ---------- -------- -----------

CANDLE1.KTS12C22                      3,384      3,384    100.00%       28
CANDLE1.KTS12C33                      3,384      3,384    100.00%       28
CANDLE1.KTS12C44                      3,345          0                 125

 DROPPED ROWS     5

** TOTALS **      6                 10,329      6,972     67.50%       186

-------------------------------------------------------------------------------------------------------------------------------------------------------

COMPRESS ROW USAGE
      ROWS IN SPACE:                          10,329    1
      DICTIONARY PAGES:                           16    2
      FREE SPACE (PCTFREE):                  50%        3
      FREE PAGE (FREEPAGE):                   4         4
      PAGES WITH ESA COMPRESSION:                237    5
      EST. PAGES W/O ESA COMPRESSION:            294    6
      PERCENTAGE OF PAGES SAVED:                  19 %  7
```

Page 3 of the report shows the Dictionary Report, Compressed Rows Summary, and Compressed Space Usage reports.

These reports are printed only if ESA compression is in effect at the time of the scan.

Dictionary Report

The Dictionary Report contains the following statistics on the DB2 dictionary pages:

1. Number of dictionary pages counted after the scan
2. Number of dictionary entries—508 for 4 K pages, 4,902 for 32 K pages
3. Page size—4 K or 32 K
4. Total number of pages required to store the dictionary

Analysis

The sample shows that the DB2 dictionary uses 64 K of memory.

Compressed Rows Sum Report

The Compressed Rows Summary report contains the following usage statistics for compressed space rows:

1. Number of rows in each table
2. Number of compressed rows in each table
3. Percentage of compressed rows in each table
4. Number of occupied pages in each table
5. Number of rows from dropped tables
6. Summary totals for the entire table space

Analysis

The sample shows the following:

- No dropped tables' rows are present.
- ESA compression is in effect for 2 of the 3 tables.

Compressed Space Usage Report

The Compressed Space Usage report summarizes statistics for compressed rows. This report contains the following:

1. Total number of rows in the table space
2. Number of dictionary pages in the table space (Controls compression and decompression)
3. Percentage of pages to remain free
4. Number of pages between free pages
5. Number of pages having data rows (exclusive of dictionary pages)
6. Estimate of required pages if the table space was not compressed. The estimate is based on an average row size and could be skewed if the row contains VARCHAR data.

   If the value for Percentage of Pages Saved is negative, then this field contains the text ****NEGATIVE COMPRESSION****
7. Percentage of pages in a table space or partition that was saved by compression. The estimate is based on an average row size and could be skewed if the value in 6 is skewed.

Analysis

The sample shows the following:

- Half (50%) of the pages in this table space are free.
- Compression saves 19% of the total pages.

```
MM/DD/YY   15:33:58        ----   D B / S M U  V XXX  DB2=D31A ----                          PAGE 16

                                D B / S M U    STATISTICS REPORTS          DBD = KCS01DB     TABLESPACE = KTS12CG      PAGE 4


              UPDATED PAGES

                     DATA BASE - QGHIJKXX

                  TABLE SPACE - SPRTAE87


                 FRACTION OF ALL PAGES IN EACH 2% OF DATASET                    CUMULATIVE FRACTION OF ALL PAGES IN DATASET

       100% |                                                  |          100% |                                                  |
            |                                                  |               |                                                  |
        90% |                                                  |       ACCUM 90% |                                                  |
            |                                                  |               |                                                  |
    PCT  80% |                                                  |         PCT 80% |                                                  |
            |                                                  |               |                                                  |
     OF  70% |                                                  |          OF 70% |                                                  |
            |                                                  |               |                                                  |
    ALL  60% |                                                  |       TOTAL 60% |##################################################|
            |                                                  |               |##################################################|
  PAGES  50% |#                                                 |         UP- 50% |##################################################|
            |#                                                 |               |##################################################|
     IN  40% |#                                                 |       DATED 40% |##################################################|
            |#                                                 |               |##################################################|
   EACH  30% |#                                                 |       PAGES 30% |##################################################|
            |#                                                 |               |##################################################|
2 PCT  20% |#                                                 |          20% |##################################################|
            |#                                                 |               |##################################################|
        10% |#                                                 |          10% |##################################################|
            |#                                                 |               |##################################################|
         0% |----+----+----+----+----+----+----+----+----+----+|           0 |----+----+----+----+----+----+----+----+----+----+|
            0   .1   .2   .3   .4   .5   .6   .7   .8   .9    1              0   .1   .2   .3   .4   .5   .6   .7   .8   .9    1
              FRACTION OF DISTANCE THROUGH THE PHYSICAL DATASET                FRACTION OF DISTANCE THROUGH THE PHYSICAL DATASET
```

Page 4 of the report contains two graphs that show the distribution of updated pages in the table space. Use the following to interpret the graphs.

- Each graph is 50 columns wide.
- Each of the grid marks, labeled .1, .2, and so forth, represents 10% of the physical data set.
- Each column represents 2% of the DASD occupied by the table space. (The sample table space contains a total of 792 pages, the equivalent of 4.4 cylinders on 3390 DASD. Each column, therefore, represents approximately 16 pages, just under 1 cylinder.)

Updated Pages

The graph on the left shows the location of pages that have been updated since the last DB2 image copy. The sample shows that 50% of the pages in the first 2% of the data set have been updated. This table space is not a candidate for an image copy.

Cumulative Updated Pages

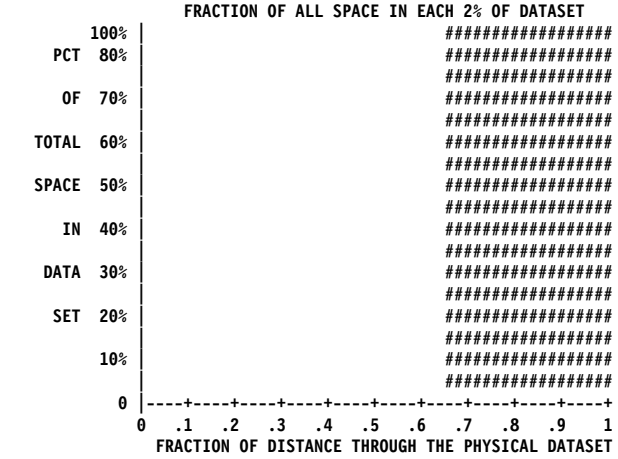
The graph on the right is a *cumulative* graph for updated pages. The graph shows at each position through the data set what percentage of the total updated pages exist up to that point.

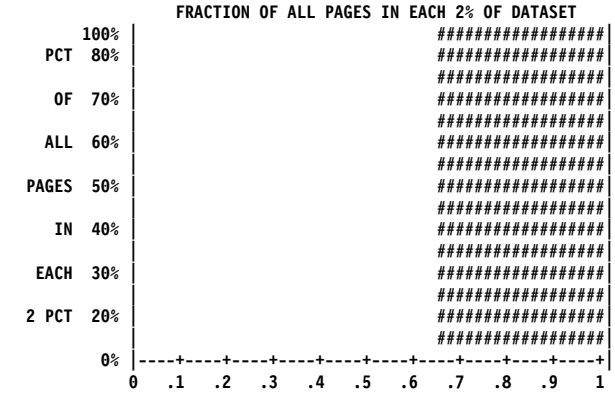
```
09/08/99   15:33:58        ----   D B / S M U  V 500  DB2=D31A ----                    PAGE 16
```

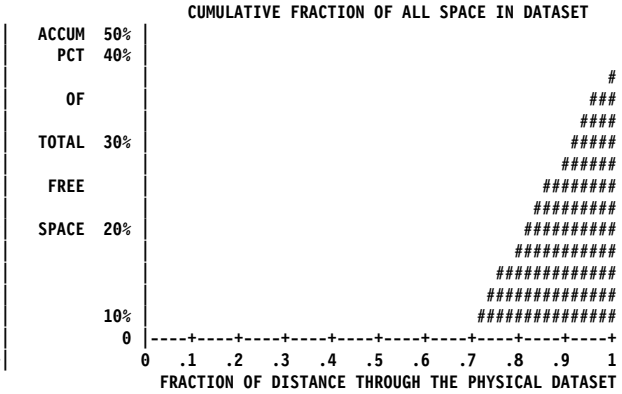
FREESPACE

DATA BASE - QGHIJKXX

TABLE SPACE - SPRTAE87







Page 5 of the report contains three graphs that show the distribution of free space in a table space. Use the following to interpret the graphs.

- Each graph is 50 columns wide.
- Each of the grid marks, labeled .1, .2, and so forth, represents 10% of the physical data set.
- Each column represents 2% of the DASD occupied by the table space. (The sample table space contains a total of 792 pages, the equivalent of 4.4 cylinders on 3390 DASD. Each column, therefore, represents approximately 16 pages, just under 1 cylinder.)

These graphs are unrelated to the earlier table space reports. They were generated for illustration purposes on an table space that contained sufficient data to generate the reports. The graphs have been squeezed at the top and bottom to fit on the page.

Distribution of Pages with Free Space Graph

Shows the percentage and location of pages that contain enough free space for at least 1 new row. In the sample, the first 64% of the pages have no room for updates; all the pages in the last 36% of the data set are free and can accept updates.

Distribution of Free Space Graph

Shows the percentage and location of available free space. A column at 30% indicates that 30% of the space within that section of the table space is free space.

Cumulative Distribution of Free Space Graph

Each column shows the cumulative percentage of free space up to that point in the data set. In the sample, the last column shows that 36% of the table space is free.

```
MM/DD/YY   15:33:58        ----   D B / S M U  V XXX  DB2=D31A ----                          PAGE 4

                              D B / S M U    STATISTICS REPORTS         DBD = QGHIJKXX    INDEXSPACE = IRTAES87      PAGE 6

                                   **************************************************
                                   *                                                *
                                   *       DBD = QGHIJKXX    INDEXSPACE = IRTAES87    *
                                   *                                                *
                                   **************************************************
 INDEX KEYS SUMMARY


 1          KEY LENGTH            8
                                                NON-UNIQUE KEYS              7
 2      RECORDS INDEXED       396,225
 3        DISTINCT KEYS       204,333          1.9  AVG RECORDS PER KEY        8
 4           LEAF PAGES         2,093        189.3  AVG RECORDS PER PAGE       9

 5  LEVELS IN THE INDEX             3
 6 ESTIMATED AFTER REORG            3             PCT PAGES USED  = 93.6        10

                                                  PCT FREESPACE = 48.0        11

    ---------------------------------------------------------------------------------------------------------------------------------

 PAGES SUMMARY
                                    NUMBER         DASD       PCT OF
                                  OF PAGES        SPACE        TOTAL
                                 -----------     --------     --------
                  CONTROL PAGES           2           8K        0.09%   1
                    USED PAGES        2,105       8,420K       93.56%   2
                   EMPTY PAGES          142         568K        6.31%   3
              UNFORMATTED PAGES            1           4K        0.04%   4

                          TOTAL        2,250       9,000K      100.00%   5
   ESTIMATED PAGES AFTER REORG        1,298   6

            FREESPACE IN DATA                     4,320K       48.00%   7
              TOTAL FREESPACE                     4,324K       48.05%   8
          FREESPACE SPECIFIED 10%  9
           FREEPAGE SPECIFIED 0%  10
```

Page 6 of the report shows the Index Keys Summary and the Pages Summary reports.

Index Keys Summary Report

The Index Keys Summary report matches the online version. It contains the following statistics that summarize the overall characteristics of the index:

1 Sum of the lengths of all key columns, plus 2 for each VARCHAR column, which are counted at full length, plus 1 for each nullable column

2 Number of indexed records

3 Number of distinct keys

4 Number of leaf pages

5 Number of levels currently in the index

6 Number of levels estimated for this index if reorg is run immediately after the scan (based on the free space and free page specifications). Variable length rows can skew the results.

7 Key type:

- Non-Unique
- Unique
- Unique primary (Indicates the primary index in a relationship involving referential integrity)

8 Average number of records per key (For unique indexes, this number is 1.0.)

9 Average number of records per page

10 Percentage of used pages in the index

11 Percentage of free space in the index space (Includes free space in unused pages and used pages)

Analysis

The sample shows the following:

- This is a non-unique index. (It cannot be unique because the number of indexed records (396,225) differs from the number of distinct keys (204,333).

Pages Summary Report

The Pages Summary report matches the online version and shows the allocation of pages in the index. It contains the following summary statistics for index subpages:

1 Number of DB2 control pages

2 Number of used pages

3 Number of empty, formatted pages no longer holding data and the number of unused, formatted pages

4 Number of empty, unformatted pages never used by DB2

5 Totals for pages, DASD space, and percentage

6 Number of pages estimated for this index space if Reorg is run immediately after the scan

7 Percentage of free space in the used data pages

8 Percentage of total free space in the entire index space (Incluedes the free space in the data pages and the empty and unformatted pages)

9 Percentage of specified free space (This value is specified in the FREESPACE parameter of the CREATE INDEX statement.)

10 Distribution of free pages. Every 10th page is empty. (This value is specified in the FREEPAGE parameter of the CREATE INDEX statement.)

Analysis

The sample shows the following:

- The high value for Empty Pages indicates significant insert activity in the index and suggests the need to reorganize the index.
- Freespace in Data is the amount of free space in the used index pages.

```
MM/DD/YY   15:33:58        ----   D B / S M U  V XXX  DB2=D31A ----                              PAGE 7

                                  D B / S M U    STATISTICS REPORTS         DBD = QGHIJKXX    INDEXSPACE = IRTAEM87      PAGE 7


 INDEX LEVELS SUMMARY

  1     2            3            4          5           6            7           8
 IX   PAGES IN      KEYS IN        KEYS    KEYS HAVE               AVG DIST    AVG DIST
 LVL    LEVEL         LEVEL       /PAGE     ROOM FOR    FREESPACE  IN PAGES     IN TRKS
 ---  ---------  -----------   --------  -----------  ---------  --------    --------
   1         1             2        2.0          808       100%

   2         2           929      464.5          691        43%       1.0           0

 LEAF      929        11,168        0.2      172,226        55%       3.8         0.4

           CLUSTERRATIO  80
    --------------------------------------------------------------------------------------------------------------------------------

 SUBPAGES SUMMARY

        SUBPAGES PER PAGE            4  1
     SIZE OF EACH SUBPAGE          993  2

            DISTINCT KEYS          168  3 0.1    AVG KEYS PER SUBPAGE  5
          RECORDS INDEXED      396,225  4 189.0  AVG RECS PER SUBPAGE  6

          TOTAL NO. SUBPAGES     3,716  7
            USED SUBPAGES        2,096  8 56.40%
           EMPTY SUBPAGES        1,620  9 43.60%
```

Page 7 of the report shows the Index Level Summary and the Subpages Summary reports.

Index Level Summary Report

The Index Level Summary report is the same as the online version. It contains the following summary statistics for pages, keys, and free space:

1. Number of the index level
2. Number of pages in the index level
3. Number of keys in the index level
4. Number of keys in the page (The number of keys should always equal the number of pages at the next lower level.)
5. Number of additional keys the level can support (Indicates the likelihood of an index level split.)
6. Percentage of free space in the level
7. Average number of pages between pages in the level—for the leaf level, this value is computed from logically consecutive pages. (Indicates the need to reorganize the index space)
8. Average number of tracks between pages in the level—for the leaf level, this value is computed from logically consecutive pages. (Indicates the need to reorganize the index space)
9. Measurement of how well the physical ordering of the table space matches the logical ordering of the keys in its index (A value of 100 indicates a perfect match.)

Analysis

The sample shows the Avg Dist in Pages (3.8) for the leaf level. This indicates a need to reorganize the index because 3.8 pages physically separate each leaf page. Reorganizing the index changes this value to 1.

Subpages Summary Report

The Subpages Summary report analyzes the use of subpages in the index. This report contains the following information:

1. Number of subpages per index page
2. Size of each subpage (Overhead in the physical index space makes the *actual* size of the subpage less than 4K.)
3. Number of distinct keys in the index
4. Number of records indexed
5. Average number of keys per subpage
6. Average number of records per subpage
7. Total number of subpages
8. Quantity of used subpages
9. Quantity of empty subpages

Analysis

This report shows the number of keys and records per subpage. The high value for keys per record indicates that 4 subpages per page might waste DASD. This index should probably be recreated with one subpage per page. Subpages reduce index lockout, but any value greater than 1 for subpages also increases DASD usage. However, having fewer locked keys improves performance and can compensate for the extra DASD.

```
MM/DD/YY   15:33:58        ----   D B / S M U  V XXX  DB2=D31A ----                          PAGE 8

                                D B / S M U    STATISTICS REPORTS          DBD = QGHIJKXX    INDEXSPACE = IRTAEM87      PAGE 8


 KEYS DISTRIBUTIONS

DISTRIBUTION OF PAGES WITH ROOM FOR KEYS                          DISTRIBUTION OF RECORDS PER KEY

  NO. KEYS WILL FIT IN "N" SBPGS       PCT OF TOTAL SUBPAGES     |   NO. RECS INDEXED BY "N" KEYS         PCT OF TOTAL KEYS
                                                                 |
     0-0                 1      0% |                           |      0-0                 0      0% |
     1-1                 1      0% |                           |      1-1                 0      0% |
     2-5               703     18.9% |#####                      |      2-5                 0      0% |
     6-10                8      0.2% |                           |      6-10                0      0% |
    11-15              149      4.0% |#                          |     11-50                0      0% |
    16-20              883     23.8% |######                     |     51-100               0      0% |
    21-25              113      3.0% |#                          |    101-250               2      1.2% |
    26-50               10      0.3% |                           |    251-500             156     92.9% |######################
    51-100           1,848     49.7% |############               |    501-1000              0      0% |
   101-                  0      0% |                           |   1001-                 10      6.0% |#
                               |----+----+----+----+----+       |                               |----+----+----+----+----+
TOTAL NO. SUBPAGES   3,716          0   20   40   60   80  100 | TOTAL NO. KEYS          168          0   20   40   60   80  100
                                                                 |

    ---------------------------------------------------------------------------------------------------------------------------------

TEN MOST FREQUENT KEYS

       NO. RECS    KEY VALUE          (FIRST 16 BYTES)
        INDEXED    CHARACTER          HEX      4        8        C
        -------    ----------------   ------------------------------------
        35,851     ..                 8009
        35,397     ..                 8003
        35,238     ..                 8001
        34,533     ..                 8006
        33,597     ..                 800A
        33,308     ..                 8002
        31,877     ..                 8005
        31,680     ..                 8004
        29,400     ..                 8007
        28,316     ..                 8008
```

Page 8 of the report contains three graphs on keys distribution: the Distribution of Pages with Room for Keys, the Distribution of Records per Key, and the Ten Most Frequent Keys graphs.

Distribution of Pages with Room for Keys Graph

The Distribution of Pages with Room for Keys graph is the same as the online version. The graph indicates how full the index pages are.

Analysis

Line three (2–5) shows that 703 subpages (18.9% of the 3,716 subpages) have free space for 2 to 5 more distinct keys.

Distribution of Records per Key Graph

The Distribution of Records per Key graph is the same as the online version. The graph shows the distribution of records per key. The number of records per key points to the even distribution of records per distinct key.

Analysis

The 8th line of the distibution (251–500) shows that 156 distinct keys (21% of the 168 distinct keys) index between 251 and 500 records each. Only 10 distinct keys, which are listed on the next graph, index more than 1000 records.

Ten Most Frequent Keys Graph

The Ten Most Frequent Keys graph is the same as the online version. This graph lists the top 10 keys, ranked by number of records indexed per key.

Analysis

The graph shows an even distribution of records per key for these 10 keys. The most frequently used key is 8009, with 35,851 records using this key. The keys are SMALLINT values, 2 bytes each.

```
MM/DD/YY   15:33:58        ----   D B / S M U  V XXX  DB2=D31A ----                            PAGE 16
                                 D B / S M U    STATISTICS REPORTS         DBD = QGHIJKXX    INDEXSPACE = IRTAEM87      PAGE 9


   INDEX USAGE GRAPH                                                  INDEX FREESPACE GRAPH


      100% | #############################################    |         100% |                                               ###
           | #############################################    |              |                                               ###
       90% |##############################################    |          90% |                                               ###
           |##############################################    |              |                                               ###
  PCT  80% |##############################################    |     PCT  80% |                                               ###
           |##############################################    |              |                                               ###
   OF  70% |###############################################   |      OF  70% |                                              ####
           |###############################################   |              |#                                             ####
  ALL  60% |###############################################   |     ALL  60% |#  #    #######      ## ## #   ## ####      ######
           |###############################################   |              |##################################################
PAGES  50% |###############################################   |   SPACE  50% |##################################################
           |###############################################   |              |##################################################
   IN  40% |###############################################   |      IN  40% |##################################################
           |###############################################   |              |##################################################
 EACH  30% |###############################################   |    EACH  30% |##################################################
           |###############################################   |              |##################################################
2 PCT  20% |###############################################   |   2 PCT  20% |##################################################
           |###############################################   |              |##################################################
       10% |###############################################   |          10% |##################################################
           |###############################################   |              |##################################################
        0% |----+----+----+----+----+----+----+----+----+----+|           0% |----+----+----+----+----+----+----+----+----+----+
           0   .1   .2   .3   .4   .5   .6   .7   .8   .9    1              0   .1   .2   .3   .4   .5   .6   .7   .8   .9    1
             FRACTION OF DISTANCE THROUGH THE PHYSICAL DATASET                 FRACTION OF DISTANCE THROUGH THE PHYSICAL DATASET
```

Page 9 of the report shows the Index Usage Graph and the Index Freespace Graph.

Index Usage Graph

The Index Usage graph is the same as the online version. It shows the distribution of used index pages in the data set. Each column represents 2% of the index.

Analysis

This graph shows the percentage of active pages in each 2% of the index. The graph shows that virtually every page in the first 95% of the index is used.

Index Free Space Graph

The Index Free Space graph is the same as the online version. This graph shows the distribution of free space in the index.

Analysis

The even distribution in this graph shows that this index was built by DB2 INSERT operations. When records are inserted into an index, the index pages split in two with each holding half of the original data. This index may need reorganization.

Example of a Summary Report for a Full Batch Scan

Overview

A full batch scan automatically generates a summary report. This unit contains an example of this report.

Purpose

The report gives a 1-line summary of each DB2 dataset that was processed in the batch job. It presents an overview of all the table spaces and index spaces in the profile and serves as a high-level gauge of DASD usage.

Elements of a summary report for a full batch scan

The summary report for a full scan contains one line for each DB2 space that was scanned. The following figure is an example.

```
 MM/DD/YY  15:33:58        ----   D B / S M U  V 500  DB2=D31A ----                          PAGE 16

                                      ******   D B / S M U   SUMMARY  *****


RUN             TABLE   INDEX    TOTAL     PCT     TOTAL  DASD     PCT   COPY REORG   ERROR
PROFILE  DBDNAME SPACE   SPACE    PAGES     USED     ROWS XTNTS  FREESP FACTOR FACTOR  PAGES
  1       2      3       4        5        6        7      8      9      10     11      12
------- -------- ------- -------- ------- ------  ------- -----  ------- ------ ------  -----

MANUAL  QGHIJKXX SPRTAE87         7,200    64.0%  396,225     1   36.0%   64.0   64.0 0

                         IRTAEM87 1,000    93.9%  396,225    11   58.0%               0

                         IRTAES87 2,250   100.0%  396,225    4    48.0%               0

222 DATASET MIGRATED SO NOT PROCESSED             DSNCAT.DSNDBD.QGHIJKXX.IRTASR87.I0001.A001  13

222 DATASET MIGRATED SO NOT PROCESSED             DSNCAT.DSNDBD.QGHIJKXX.IRTASR87.I0001.A001  13

                 SPRTAE88         7,200    56.7%  359,267    2    43.3%   56.8   56.7 0

                         IRTAEM88   900    95.4%  359,267          9     57.6%        0

                         IRTAES88 2,100   100.0%  359,267          5     45.5%        0

                         IRTASR88   800    87.5%  359,267          1     43.6%        0
```

Elements of a summary report for a full batch scan (continued)

1 Name of the run profile that initiated the scan

2 Name of the DB2 database

3 Name of the table space that was scanned

4 Name of the index space that was scanned

5 Total number of DB2 pages (size 4K or 32K) that are allocated to the dataset

6 Percentage of the total pages that are or have ever been used. (The others are unformatted.)

7 For table spaces, the total number of row in all the tables in the table space

For index spaces, the number of records indexed, *not* the number of distinct keys. (If there is only one table in the table space, the number of records indexed should equal the number of keys. This is a simple check on index-table consistency.)

8 Total number of DASD extents allocated to the dataset

9 Percentage of the total DASD space allocated to this dataset that is free space or unformatted space

10 Percentage of tracks that have been updated (table spaces only)

11 Weighted percentage of pages with free space (table spaces only)

12 Number of different errors detected in the dataset. (If this value is nonzero, there are corresponding messages in the ERRORS file.)

13 Message printed if you specified that migrated datasets were not to be recalled

Example of a Summary Report for a Batch Space Map Analysis

Overview

A batch space map analysis generates a summary report. This unit contains an example of this report.

Purpose

The report gives a 1-line summary for each DB2 dataset that !DB/SMU processed in the batch job. For each dataset, it provides an overview of all the table spaces and index spaces in the profile and serves as a high-level gauge of DASD usage. It also provides summary statistics for the run profile used to generate the space map analysis.

The batch space map analysis summary report consists of two parts. The first part provides space map analysis information by database for each space you identified to be analyzed. The second part of the report provides summary statistics for all spaces.

Elements of a summary report for a space map analysis (continued)

This is an example of the space map analysis summary report.

```
 MM/DD/YY   15:33:58         ----   D B / S M U  V 500 DB2=D31A ----                              PAGE 16

TABLE TABLE INDEXED                PAGES  PCT  DASD   COPY  REORG
SPACE       INDEX NAME           IN DSET USED XTNTS FACTOR FACTOR DSNAME
 2     3      4                     5    6    7     8      9     10
----- ------------------------- ------- ---- ----- ------ ------ ----------------------------------------
**DBD=DSQDBCTL ** 1

DSQTSCT1 2                           100 69.0    1      .0   35.0 DSNCAT.DSNDBD.DSQDBCTL .DSQTSCT1.I0001.A001
      Q.OBJECT_DIRECTORY 3
              Q.OBJECT_DIRECTORYX 4   60 98.4    2    98.5        DSNCAT.DSNDBD.DSQDBCTL .OBJE1PTV.I0001.A001

DSQTSCT2 2                           100 71.0    1      .0   68.0 DSNCAT.DSNDBD.DSQDBCTL .DSQTSCT2.I0001.A001
      Q.OBJECT_REMARKS 3
              Q.OBJECT_REMARKSX 4     60 98.4    2    98.5        DSNCAT.DSNDBD.DSQDBCTL .OBJE1VXT.I0001.A001

DSQTSCT3 2                         5,250 97.8   26      .0   97.5 DSNCAT.DSNDBD.DSQDBCTL .DSQTSCT3.I0001.A001
      Q.OBJECT_DATA 3
              Q.OBJECT_DATAX 4       100 66.0    1    66.0        DSNCAT.DSNDBD.DSQDBCTL .OBJE1RPK.I0001.A001

DSQTSGOV 2                            30 13.4    1    33.5    6.5 DSNCAT.DSNDBD.DSQDBCTL .DSQTSGOV.I0001.A001

DSQTSPRO 2                           100 38.0    1      .0   14.5 DSNCAT.DSNDBD.DSQDBCTL .DSQTSPRO.I0001.A001
      Q.PROFILES 3
              Q.PROFILEX 4            30 26.7    1    26.5        DSNCAT.DSNDBD.DSQDBCTL .PROFILEX.I0001.A001

DSQTSSYN 2                            30 13.4     1    33.5    6.5 DSNCA T.DSNDBD.DSQDBCTL .DSQTSSYN.I0001.A001
      Q.COMMAND_SYNONYMS 3
              Q.COMMAND_SYNONYMSX 4  30 10.0     1    10.0        DSNCAT.DSNDBD.DSQDBCTL .COMMANDR.I0001.A001


** DBD = DSQ1STBB ** 1

DSQ1STBT 2                          100  4.0    1      .0    2.0 DSNCAT.DSNDBD.DSQ1STBB .DSQ1STBT.I0001.A001
      Q.STAFF 3
              AK68CL.XID 4           10 30.0    1    30.0        DSNCAT.DSNDBD.DSQ1STBB .XID.I0001.A001
              .                       .               .
              .                       .               .
              .                       .               .
              .                       .               .
 11 PROFILE=RSQ             TBL SPACES         INDX SPACES        TOTALS
                             --------           ----------         ------
 12 NO. OF SPACES                29                 41                70

 13 NO. OF PAGES              53,706              1,112            54,818

 14 NO. OF EXTENTS               112                 66               178

 15 NO. OF TRACKS              4,948                 96             5,044

 16 NO. OF CYLINDERS             339                 41               380

 17 PAGES UPTD/USED           17,865                471            18,336

 18 PCT PGS UPD/USED          33.26%              42.36%            75.62%

 19 FREE INDEX TRKS                                 374               374

ALL UPDATES PROCESSED
```

1 Number of dataspaces

2 Name of the DB2 table space

3 Name of the DB2 table in the table space

4 Name of the index for the table

Elements of a summary report for a space map analysis (continued)

1 Number of dataspaces

2 Name of the DB2 table space

5 Total number of DB2 pages (size 4K or 32K) that are allocated to the dataset

6 Percentage of the total pages that are now used or have ever been used

7 Total number of DASD extents allocated to the dataset

8 For table spaces, the percentage of tracks that have been updated since the previous image copy

For index spaces, the percentage of index pages currently active

9 Weighted percentage of pages in a table space that have free space

10 The fully qualified dataset name of the DB2 space

11 The name of the run profile used to generate this space map analysis

12 Number of datasets processed

13 Total number of pages allocated to the databases

14 Number of extents allocated to the databases scanned during the analysis

15 Number of tracks occupied by the databases scanned during the analysis

16 Number of cylinders occupied by the databases scanned during the analysis

17 Total number of pages now used or ever used

18 Percent of allocated pages now used or ever used

19 Number of unused index tracks

Chapter 32.
Customized User-Written Reports

Introduction

This chapter contains information about producing user-written reports that tap summary data collected by !DB/SMU.

Chapter Contents

Background for Customized Reports

Overview

This unit provides information on the sources of summary data you can use to produce customized reports.

Background about gathering summary data

The summary data collected by !DB/SMU becomes an effective history of the physical characteristics of each table space and index space that the batch scan or monitor commands process.

!DB/SMU collects summary data in two ways:

When !DB/SMU ...	THEN ...
Completes processing a batch SCAN command that initiates the RUNSTATS option,	It inserts the summary data into a DB2 table called SPCSTATS.
Completes processing a batch MONITOR command,	It inserts the summary data into a flat file specified by the //MNTFILE DD statement.

Using the summary data to create reports

You can access the summary data to create custom reports in the following ways:

Source of Summary Data	Ways to Access Data
SPCSTATS DB2 table	• Use the DB2 SPUFI facility to run SQL queries against the table. • Use QMF to return answers to queries or display charts and graphs derived from the table data. • Write a custom program to read the SPCSTATS table.
Flat file produced by the batch MONITOR command	• Use the flat file as input to any custom routine.

DB2 Table for Scan RUNSTATS Data

Overview

This unit describes the DB2 table in which !DB/SMU stores summary data when you select the RUNSTATS option for a batch scan.

How !DB/SMU creates the SPCSTATS table

The SPCSTATS table is constructed during the installation process. The CREATE DDL is included in the distribution CNTL dataset. The name of the table is SPCSTATS. *If you change the table name or create a view, you must update the CLIST KTSSPCU table parameter.*

Description of the SPCSTATS table

This unit describes the data in the SPCSTATS table in terms of

- Column names
- Type of data (for example, character, decimal, integer, and so forth)
- Length in bytes
- Offset
- Description of the data

Refer to the following chart for a description of the summary data in SPCSTATS.

Description of the SPCSTATS table (continued)

Column Name	Data Type	Length	Offset (in Hex)	Description
DBNAME	CHAR	8	0	Database name
TSNAME	CHAR	8	8	Table space name
TSPART	SMALLINT	2	A	Table space partition number
IXNAME	CHAR	8	12	Index space name (if an index)
IXPART	SMALLINT	2	14	Index space partition number
SPCTYPE	CHAR	1	15	Space type **S** Table space **X** Index space
RUNTIME	TIMESTAMP	10	1E	Timestamp entry added to table
TOTAL_PAGES	INTEGER	4	23	Number of pages in the space
TRKS	INTEGER	4	27	Tracks allocated
CYLS	INTEGER	4	2A	Cylinders allocated
UPDATED_PAGES:	INTEGER	4	2E	Number of pages updated since previous image copy
USED_PAGES	INTEGER	4	33	Number of index pages in use
EXTENTS	SMALLINT	2	35	Number of DASD extents
REORG_FACTOR	DECIMAL	5 / 1	3A	Reorg factor value (0–100)
COPY_FACTOR	DECIMAL	5 / 1	3E	Copy factor value (0–100)

Optional Exception File for Monitor Data

Overview

This unit describes the optional exception file that is created as a byproduct of processing a batch MONITOR command.

Output of the batch MONITOR command

When you execute the MONITOR command during a batch run, !DB/SMU creates the following types of output:

- Monitor exception reports

  To produce these reports, you must first create a monitor run profile using the !DB/SMU panels. See "Defining and Using Run Profiles for Batch Jobs" on page 491 for more detail on the monitor run profile.

- An optional file updated with summary values that you can use to produce customized user-written reports.

  In order to create this optional file, you *must add* an //MNTFILE DD statement to the JCL !DB/SMU creates to run its batch utility.

How !DB/SMU creates the file

The optional exceptions file is produced by the //MNTFILE DD statement in the JCL that !DB/SMU builds for the batch job. The file contains 1 record for each table space and index space that satisfies the criteria in the monitor run profile.

The file can be used as input to any installation-written routines to process the selected datasets.

Description of the file for MONITOR data

This chart describes the columns of the DB2 file for monitor exceptions summary data.

Data Type	Length	Offset (In Hex)	Description
CHAR	6	0	Profile name
CHAR	1	6	Space type **S** Table space **X** Index space
CHAR	1	7	Size unit. **K** K-bytes **M** M-bytes **C** Cylinders **T** Tracks **P** Pages
CHAR	8	8	Database name
CHAR	8	10	Table space name
CHAR	2	18	Table space partition number (character)
CHAR	8	1A	Table space creator
CHAR	8	22	Index space name
CHAR	2	2A	Index space partition number
CHAR	8	2C	VSAM catalog name
CHAR	18	34	Index name
CHAR	8	46	Index creator
BINARY	4	50	Data set size
BINARY	2	54	Number of DASD extents

Description of the file for MONITOR data (continued)

Data Type	Length	Offset (in Hex)	Description
BINARY	2	56	Copy factor. The copy factor field contains a value from 0–200. The copy factor is computed from this field by dividing the value by 2. For example, if the field value is 155, the copy factor is 155/2 = 77.5.
BINARY	2	58	Reorg factor. The reorg factor field contains a value from 0–200. The reorg factor is computed from this field by subtracting the value from 200 and then dividing by 2. For example, if the field value is 155, the reorg factor is (200–155)/2 = 22.5.
CHAR	8	5A	Date the space was analyzed (*yy/mm/dd*)
CHAR	8	62	Run date of the MONITOR command (*yy/mm/dd*)
CHAR	8	6A	Job name of the MONITOR command

Appendixes

Appendix A. Batch Commands You Can Enter Manually

When you have generated the JCL to run the !DB/SMU batch utility, you can manually add commands to the JCL if you wish. The applicable commands are

- DOCOMPLETEJOBS
- DUMP
- FORMAT
- STACK

This appendix provides you with the command syntax, an example of the use of the command syntax, and, for DUMP and FORMAT, examples of the output of the command's execution.

Appendix Contents

Using the DOCOMPLETEJOBS Command in a Batch Job

Overview

This unit contains instructions and examples for using the DOCOMPLETEJOBS command.

Use

The DOCOMPLETEJOBS command causes !DB/SMU to bypass checking to determine whether your batch JCL generates utilities that might be considered ambiguous.

Under normal processing conditions, !DB/SMU will cancel ambiguous utility requests. Examples of such requests are:

- A a table space REORG followed by a request to REORG the associated index spaces.
- RUNSTATS on a table space with the option INDEX(ALL) followed by a request for RUNSTATS on an associated index space.

In both cases, the second utility generation request is redundant. Including the command DOCOMPLETEJOBS in your batch SCAN or MONITOR job stream, just ahead of the SCAN or MONITOR keyword, will cause !DB/SMU to bypass checking for the ambiguous generation request.

Syntax example

This example uses the utility profile TESTR and scan profile BATSCAN and illustrates the position of the DOCOMPLETEJOBS command within the batch job stream.

```
//A       EXEC BUTILITY
//SYSTSIN   DD *
 ISPSTART CMD(%KTSXSMU DB2(DB31)) NEWAPPL(KTC1)
//SYSIN     DD *
 UTIL OPTS  TESTR
 DOCOMPLETEJOBS
 SCAN      BATSCAN
/*
```

Using the DUMP Command in a Batch Job

Overview

This unit contains instructions and examples for using the DUMP command.

Use

The DUMP command reports the hexadecimal values in a range of DB2 pages for a table space or an index space.

Command syntax for the DUMP command

You manually add the DUMP command to a generated job stream for any run profile. Apply the rules for batch commands given in "Commands for Batch Jobs" on page 482. Begin the command in any column, and use the following syntax for the DUMP command.

```
DUMP  DB(name) IX(name) | TS(name)  FROMPAGE(n) | TOPAGE(n)
```

or

```
DUMP  DB(name) IX(name) | TS(name)  FP(n) | TP(n)
```

Abbreviations for the DUMP command

DU and D are valid abbreviations.

Example of a DUMP command

The following example illustrates the use of the DUMP command:

```
DUMP DB(DBTX1) TS(STAFF) FP(00) TP(33)
```

Parameters for the DUMP command

The following chart describes command parameters and the modifications they require. All the parameters are required unless otherwise indicated.

Parameter	Value You Supply
DB(*name*)	Name of the database that contains the table space or index space to be dumped (Optional if the table space name or index space name is unique)
TS(*name*) | IX(*name*)	Name of the table space or index space to be dumped. Specify the space name without its leading qualifier, the Creator ID.
FROMPAGE(n) or FP(n)	Starting page number of the range of pages to be dumped. Specify the page number in either decimal or hexadecimal. Leading zeros are presumed. Decimal values: Use 1–8 decimal digits. Hexadecimal values: Use 1–6 hexadecimal digits enclosed in single quotes and preceded by X. The format is `X'nnnnnn'`
TOPAGE(n) or TP(n)	Ending page number of the range of pages to be dumped. All the guidelines for FROMPAGE apply equally to TOPAGE.

Results of using the DUMP command

DUMP opens the specified DB2 space and dumps the requested pages.

Example job for the DUMP command

The following example adds the DUMP command to the generated job stream for a space run profile for space map analysis. This procedure is basically the same for a space map, monitor, or scan run profile.

Step	Action
1	On the Batch Run Profile Select Menu, type **1** in the Select Profile Type field.
2	Press Enter. **Result**: !DB/SMU displays the Space Profiles panel, which lists the existing profiles.
3	Identify a profile you want to use by typing **U** in the Sel field next to its name.
4	Press Enter. **Result**: !DB/SMU displays the Generate Batch Job panel.
5	Review the values on the Generate Batch Job panel and change them if needed.
6	Press Enter. **Result**: !DB/SMU generates the job statements and displays the Process Member panel.
7	On the Process Member panel, select the EDIT option by typing **2** in the Select Action field.
8	Press Enter. **Result**: The JCL you created is displayed. You can use the ISPF editor to make any required changes.

9	Scroll to the //SYSIN DD * statement at the end of the data. Delete any text that follows it, and add the DUMP command as shown in the following example.
	
	`//SYSIN DD *`
	`  DUMP DB(db_name) TS(space_name) FP(00) TP(33)`

10	Type **END** on the command line or press the appropriate function key. **Result**: !DB/SMU returns to the Process Member panel.
11	On the Process Member panel, select the UPDATE NOW option by typing **1** in the Select Action field.
12	Press Enter. **Result**: !DB/SMU submits the job to dump the specified objects.

Example DUMP output

The following figure contains example output for the DUMP command. The example contains data for pages 0–3; repeating lines are not printed in this output.

```
DATASET DSNH.DSNDBD.DSQDBCTL.DSQTSCTL.I0001.A001,   DB2 PAGE X'000000' (0) 1

00000000  00000000 0941D701  00000018 01000002   00000001 00180000  00000...
00000020  00000000 00000000  00000000 00000000   00000000 00000000  00000... 2
OFFSETS 00000040 THROUGH 00000FC0 SAME AS ABOVE
00000FE0  00000000 00000000  00000000 00000000   00000000 00000000  00000...


DATASET DSNH.DSNDBD.DSQDBCTL.DSQTSCTL.I0001.A001,   DB2 PAGE X'000001' (1)

00000000  10000000 2A0BF201  00000110 00002A08   00000005 00000000  0000 ...
00000020  00000000 00000000  00000000 00000000   00000000 00000000  0000
OFFSETS 00000040 THROUGH 00000A60 SAME AS ABOVE
00000A80  00000000 00000000  00000000 00000000   00000000 00000000  0000 ...
00000AA0  00000000 00000000  00000000 00000000   00000000 00000000  0000 ...
OFFSETS 00000AC0 THROUGH 00000FC0 SAME AS ABOVE
00000FE0  00000000 00000000  00000000 00000000   00000000 00000000  0000 ...


DATASET DSNH.DSNDBD.DSQDBCTL.DSQTSCTL.I0001.A001,   DB2 PAGE X'000002' (2)

00000000  10000000 29C26A01  00000202 06B00BF2   094C0100 00093800  0301 ...
00000020  40400006 D8E4C5D9  E8F18001 E2D8D340   40D5D640 40404040  4040 ...
00000040  404040D9 C5D340F1  4BF04000 01FF08FC   00380300 00000012  D8E4 ...
00000060  40404040 40404040  40400000 00060000   00010001 00000000  0000 ...
```

Each page in the dump provides the following information:

1 A heading that contains the

- Name of the DB2 dataset
- Page number in both hexadecimal and decimal format. (The DB2 page number is relative to 0 because the first page of the space is page 0.)

2 The hexadecimal offset into the page and the page data

Using the FORMAT Command in a Batch Job

Overview

This unit contains instructions and examples for using the FORMAT command.

Use

The FORMAT command decodes page data and produces a formatted output for a range of DB2 pages from a table space or an index space.

Command syntax for the FORMAT command

You manually add the FORMAT command to a generated job stream for a run profile. Apply the rules for batch commands given in "Commands for Batch Jobs" on page 482. Begin the command in any column, and use the following syntax for the FORMAT command.

```
FORMAT  DB(name) IX(name) | TS(name)  FROMPAGE(n) | TOPAGE(n)
```

or

```
FORMT  DB(name) IX(name) | TS(name)  FP(n) | TP(n)
```

Abbreviations for the FORMAT command

FRMT, FMT, and F are valid abbreviations.

Example of a FORMAT command

The following example illustrates the use of the FORMAT command:

```
FORMAT DB(DBTX1) TS(STAFF) FP(00) TP(33)
```

Parameters for the FORMAT command

The following chart describes command parameters and the modifications they require. All the parameters are required unless otherwise indicated.

Parameter	Value You Supply
DB(*name*)	Name of the database that contains the table space or index space to be formatted. (Optional if the table space name or index space name is unique)
TS(*name*) | IX(*name*)	Name of the table space or index space to be formatted. Specify the space name without its leading qualifier, the Creator ID.
FROMPAGE(N) or FP(n)	Starting page number of the range of pages to be formatted. Specify the page number in either decimal or hexadecimal. Leading zeros are presumed. Decimal values: Use 1–8 decimal digits. Hexadecimal values: Use 1–6 hexadecimal digits enclosed in single quotes and preceded by X. The format is **`X'nnnnnn'`**.
TOPAGE(n) or TP(n)	Ending page number of the range of pages to be dumped. All the guidelines for FROMPAGE apply equally to TOPAGE.

Results of using the FORMAT command

FORMAT opens the specified DB2 space and formats the requested pages.

Example job for the FORMAT command

The following example adds the FORMAT command to the generated job stream for a scan run profile. The procedure is basically the same for a space, monitor, or scan profile.

Step	Action
1	On the Batch Run Profile Select Menu, type **1** in the Select Profile Type field.
2	Press Enter. **Result**: !DB/SMU displays the Space Profiles panel, which lists the existing profiles.
3	Identify a profile you want to use by typing **U** in the Sel field next to its name.
4	Press Enter. **Result**: !DB/SMU displays the Generate Batch Job panel.
5	Review the values on the Generate Batch Job panel and change them if needed.
6	Press Enter. **Result**: !DB/SMU generates the job statements and displays the Process Member panel.
7	On the Process Member panel, select the EDIT option by typing **2** in the Select Action field.
8	Press Enter. **Result**: The JCL you created is displayed. You can use the ISPF editor to make any required changes.

9	Scroll to the //SYSIN DD * statement at the end of the data. Delete any text that follows it, and add the DUMP command as shown in the following example.
	`//SYSIN DD *`
	`  FORMAT DB(db_name) TS(space_name) FP(00) TP(33)`

10	Type **END** on the command line or press the appropriate function key. **Result**: !DB/SMU returns to the Process Member panel.
11	On the Process Member panel, select the UPDATE NOW option by typing **1** in the Select Action field.
12	Press Enter. **Result**: !DB/SMU submits the job to dump the specified objects.

Example output for the FORMAT command

The following figure contains example output produced by the FORMAT command.

```
FORMAT DB(KCS01DB) TS(KTS12SCG FP(0) TP (33)
                                                        *** BROKEN PAGE **
02/24/99   13:06:05        ----   D B / S M U   - DB2=D31A ----

DSN=TDDB231A.DSNDBD.KCS01DB.KTS12CG.I0001.A001

                              U  MAP        FREE
   PAGE NUMBER      LOG RBA   D  IDS  ROWS  SPACE
      1               2      3  4    5     6          7
-----------------  -------    -  ---  ----  -----
         0       0   52B2B2A8                       ***  HEADER PAGE  ***
         1       1   52B96E5*  Y                    ***  SPACE MAP PAGE ***
         2       2   52B129E5                       ***DICTIONARY PAGE**
         3       3   52B13A3A                       ***DICTIONARY PAGE**
         .       .      .                                  .
         .       .      .                                  .
         .       .      .                                  .
        17      11   52B21EE0                       ***DICTIONARY PAGE**
        18      12   52B23000      0     1   99%
        19      13   52B24055      0     1   99%
        20      14   52B250AA      0     1   99%
        21      15   52B260FF      0     1   99%
         .       .      .
         .       .      .
         .       .      .
        27      1B          0*   126   126   50%
        28      1C          0*   127   127   51%
        29      1D          0*   127   127   50%
        30      1E          0*     0     1   99%
        31      1F          0*   125   125   50%
        32      20          0*   127   127   51%
        33      21          0*   127   127   50%
```

1 Number of the page, relative to 0, given in decimal and hexadecimal format

2 DB2 relative byte address in the log at the last update of this page. (An asterisk (*) indicates an invalid log RBA.)

3 Update flag. (Y indicates that updates have been made to the page since the previous image copy.)

4 Number of map ID entries in this page

5 Number of table rows in this page

6 Percentage of free space in the page based on values from the page header

7 Page type

Using the STACK Command in a Batch Job

Overview

This unit contains instructions and examples for using the STACK command.

Use

The STACK command defers the initiation of a DB2 utility until the STACK limit is reached.

Command syntax for the STACK command

You manually add the STACK command to a generated job stream for scan or monitor run profiles. Enter the STACK command preceding the SCAN, SCANR, or MONITOR command in the job stream. Apply the rules for batch commands given in "Commands for Batch Jobs" on page 482. Begin the command in any column, and use the following syntax for the command:

```
STACK nn
```

Note that, unlike the use of STACK in an action profile or user action definition, the use of STACK as a manually-entered command *does not* permit an equals sign (=) between the command and its value. See "Specifying Actions Using Action Profiles and Inline User Action Definitions" on page 557 for information about using STACK with an action profile or user action definition.

Limitation on the use of STACK

If you specify STACK *nn* manually in your job stream, it is only effective if there is no STACK command in an associated action profile. Including a STACK command in an action profile associated with the SCAN, SCANR, or MONITOR run overrides the effect of a STACK command you have manually entered in the job stream.

Value for the STACK command

When used with the STACK command, *nn* can be a value from 1 through 32,760. If you do not specify STACK and are executing a scan, STACK defaults to 1. If you are executing a monitor run, STACK defaults to processing *all* of the objects as a single group.

Results of using the STACK command

STACK controls the number of utility jobs generated.

Example for the STACK command

The following example adds the STACK command to the generated job stream for a scan run profile. This procedure is basically the same for a monitor run profile.

Step	Action
1	On the Batch Run Profile Select Menu, type **3** in the Select Profile Type field.
2	Press Enter. **Result**: !DB/SMU displays the Scan Profiles panel, which lists the existing profiles.
3	Identify a profile you want to use by typing **U** in the Sel field next to its name.
4	Press Enter. **Result**: !DB/SMU displays the Generate Batch Job panel.
5	Review the values on the Generate Batch Job panel and change them if needed.
6	Press Enter. **Result**: !DB/SMU generates the job statements and displays the Process Member panel.
7	On the Process Member panel, select the EDIT option by typing **2** in the Select Action field.
8	Press Enter. **Result**: The JCL you created is displayed. You can use the ISPF editor to make any required changes.

Example for the STACK command (continued)

Step	Action

9	Scroll to the //SYSIN DD * statement at the end of the data. Just ahead of the SCAN *profilename* panel, add the STACK command as shown in the following example.
	
	`//SYSIN DD *`
	`  STACK 10`
	`  SCAN MYPROFILE`

10	Type **END** on the command line or press the appropriate function key. **Result**: !DB/SMU returns to the Process Member panel.
11	On the Process Member panel, select the UPDATE NOW option by typing **1** in the Select Action field.
12	Press Enter. **Result**: !DB/SMU submits the scan job.

Commands for Working with the User PDS

Overview

This unit describes two commands that enable you to edit or browse members in the user partitioned dataset (PDS).

Background about the UEDIT and UBROWSE commands

!DB/SMU provides commands to permit you to view or edit members in your user PDS without exiting !DB/SMU or using a split screen. The commands are UBROWSE and UEDIT. To execute either of these commands, enter it on the command line of the object panel you are using.

Format of the commands

The commands UBROWSE and UEDIT take the form:

```
UBROWSE [member | member pattern]
UEDIT [member | member pattern]
```

where

- *member* specifies the name of the member in the user PDS you want to browse or edit
- *member pattern* is a mask that is used to create a list of members which !DB/SMU presents to you. You can then select the member or members you want to browse or edit. Review the information in the section "Special characters used in pattern masks" on page 84 for information about creating pattern masks.

If you omit the member name or member pattern mask, !DB/SMU presents a list of the entire contents of the user PDS from which you can select the member or members to work with.

Command for Determining Your System Level

Overview

This unit provides information to assist you in determining the level of your !DB/SMU system.

Procedure to determine the maintenance level of your system

The maintenance level of your !DB/SMU system is displayed on the !DB/SMU Primary Menu. If you want to determine the maintenance level from another !DB/SMU panel, follow the steps in this chart.

Step	Action
1	To obtain the maintenance level of the !DB/SMU system you are using, type **MAINT** on the command line of a !DB/SMU panel.
2	Press Enter. **Result:** !DB/SMU displays the maintenance level of the system you are using.

Procedure to determine the maintenance level of your !DB/Tools

The maintenance level of all of the installed !DB/Tools can be displayed by following the steps in this chart.

Step	Action
1	To obtain the maintenance level of every installed component of !DB/Tools, type **MAINT?** on the command line of a !DB/SMU panel.
2	Press Enter. **Result:** !DB/SMU displays the maintenance level of the !DB/Tools you have installed.

Appendix B. Error Messages for Integrity Checks

Introduction

!DB/SMU issues a message for any integrity check that fails. This appendix groups these messages into categories. These categories parallel the checks that were described in the unit "!DB/SMU Data Integrity Checks" on page 249.

Appendix Contents

Error Messages for Data Integrity Checks Grouped by Type of Space

Overview

This appendix contains a summary of all of the !DB/SMU error messages for data integrity checks. Detailed information about each message follows the summary. The messages are listed in numerical order by message number.

Messages for both table spaces and index spaces

- Header page checks
  - 90 DBID MISMATCH
  - 91 PSID MISMATCH
  - 92 ISOBID MISMATCH
- Page header checks
  - 01 I/O ERROR FLAG ON IN HEADER
  - 02 PAGE END CHARACTER MISMATCH
  - 03 PAGE NUMBER MISMATCH
  - 04 BROKEN PAGE FLAG IS ON
  - 55 LOG RBA VALIDITY INDETERMINATE
- Page type checks
  - 08 FLAGS MISMATCH -- HEADER PAGE FLAG NOT ON
  - 09 FLAGS MISMATCH -- SPACE MAP PAGE FLAG INCORRECTLY ON
  - 10 FLAGS MISMATCH -- SPACE MAP PAGE FLAG NOT ON
  - 11 SPACE MAP COUNT INVALID
  - 19 EMBEDDED ZERO PAGE
- Processing abend
  - 99 ABEND PROCESSING PAGE

Messages for table spaces

- Page type checks
  - 05 FLAGS MISMATCH -- NOT AN INDEX
  - 56 SEGMENTED FLAG INCORRECTLY ON
- Data page checks
  - 12 FREESPACE VALUE INCORRECT, PAST MAP START
  - 13 NUMBER OF ID MAP ENTRIES = 0, OR > 127.
  - 14 LARGE HOLE FOUND NOT ON HOLE CHAIN
  - 15 ERROR IN HOLE CHAIN
  - 16 FREESPACE VALUE NOT = TOTAL FREESPACE
  - 57 UPDATE STATUS INCONSISTENCY
  - 58 UPDATE STATUS INCONSISTENCY
  - 59 INVALID OFFSET TO CONTIGUOUS FREE SPACE
- Data page record checks
  - 17 INVALID RECORD TYPE
  - 18 RECORD LENGTH EXTENDS INTO FREESPACE
- Pointer records
  - 20 DUPLICATE POINTER RECORD
  - 21 POINTER REC DOES NOT POINT TO AN OVERFLOW RECORD
  - 22 OVERFLOW REC ORPHANED -- NO MATCHING POINTER REC
- Data page ID map checks
  - 23 INVALID ID MAP ENTRY NUMBER
  - 24 ID MAP FREE CHAIN INVALID -- MISSING FREE ENTRIES
  - 25 ORPHAN ID MAP ENTRY -- NO RECORD FOR ENTRY
  - 26 ID MAP FREE CHAIN INVALID -- FREE ENTRY NOT ON CHAIN
  - 27 ID MAP FREE CHAIN INVALID -- POINTS TO NON-FREE ENTRY
  - 28 ID MAP FREE CHAIN INVALID -- LOOP IN CHAIN

Messages for segmented table spaces

- Space map page: page header and space prefix checks
  - 60 UNEXPECTED PAGE-TYPE FLAGS
  - 61 SEGMENT SIZE MISMATCH
  - 62 INVALID SEGMENT SIZE
  - 63 INVALID COVERAGE
  - 64 SIZE - COVERAGE MISMATCH
  - 65 INVALID FREE-SEGMENT COUNT
  - 66 INVALID “NEXT SEGMENT”
  - 67 INVALID LAST FORMATTED PAGE
- Space map page: segment prefix checks
  - 68 UNDEFINED SEGMENT ENTRY FLAGS
  - 69 FREE SEGMENT ENTRY IS “FIRST”
  - 70 FREE SEGMENT ENTRY HAS “NEXT”
  - 71 FREE SEGMENT ENTRY HAS OBID
  - 72 FREE SEGMENT ENTRY HAS SPACE INFO
  - 73 ZERO OBID
  - 74 MULTIPLE “FIRST” SEGMENT ENTRIES
  - 75 SEGMENT ENTRY CHAIN IS BROKEN
  - 76 LOOP IN SEGMENT ENTRY CHAIN
  - 77 FREE SEGMENT ENTRY ON CHAIN
  - 78 WRONG OBID ON CHAIN
  - 79 ORPHANED SEGMENT ENTRY
- Space map page: checks
  - 80 NON-ZERO SECTION
- Data page checks
  - 81 OBID INVALID FOR SEGMENT
  - 82 DATA PAGE BEYOND RANGE OF SPACE MAP
  - 83 SEGMENTED SPACE MAP PAGE FLAG INCORRECTLY ON

Messages for index spaces

- Page type checks
  - 06 FLAGS MISMATCH -- INDEX FLAG NOT ON
- Checks on all pages
  - 29 KEYLENGTHS MISMATCH IN PAGE
  - 30 KEYLENGTH LARGER THAN 255

Messages for index spaces (continued)

- Logical page header checks
  - 31 NO. SUBPAGES INVALID IN NON-SEGMENTED MODE
  - 32 NO. SUBPAGES HAS AN INVALID VALUE
  - 33 AT LEAST ONE SUBPAGE IS NOT EMPTY IN EMPTY PAGE
  - 34 NON-LEAF PAGES CANNOT BE SEGMENTED -- FLAG IS ON
  - 35 SUBPAGE IS MARKED BROKEN
  - 36 KEY OFFSET VALUE IS INVALID
  - 37 FREESPACE OFFSET DOES NOT POINT TO FREESPACE
  - 38 FREESPACE VALUE DOES NOT MATCH COMPUTED FREESPACE
  - 39 KEYS ARE NOT IN ASCENDING SEQUENCE
  - 40 KEY ENTRY LENGTH ERROR -- NOT MATCH PREFIX DATA
  - 41 SUBPAGES ARE NOT ALL THE SAME SIZE
  - 42 FIRST SUBPAGE DOES NOT FOLLOW SUBPAGE DIRECTORY
- Cross-page checks
  - 43 UNUSED INDEX PAGE REFERENCED BY KEY ENTRY
  - 44 KEY ENTRY POINTS TO AN UNUSED INDEX PAGE
  - 45 INDEX PAGE DOUBLY REFERENCED
  - 46 PAGES PAST END OF INDEX ARE REFERENCED PREVIOUSLY
  - 47 INDEX PAGE ORPHANED -- NOT REFERENCED BY ANY ENTRY
  - 48 NO. OF INDEX LEVELS DIFFERENT FROM PRIOR VALUE
  - 49 INDEX CHAIN DOES NOT GO DOWN TO LEAF PAGE LEVEL
- Leaf page chains
  - 50 MULTIPLE “FIRST” LEAF PAGES -- PRIOR POINTER = 0
  - 51 LOOP IN LEAF PAGE FORWARD CHAIN
  - 52 LEAF PAGE FORWARD CHAIN POINTS TO NON-LEAF PAGE
  - 53 LEAF PAGE BACK CHAIN DOES NOT POINT TO PRIOR PAGE
  - 54 ORPHANED LEAF PAGE -- NOT ON FORWARD CHAIN

Messages for index spaces (continued)

- Type 2 index checks

  Record ID (RID) chains are new with DB2 Version 4.

  - 84 CHAINED RID HAS RID HOLE FOLLOWS FLAG SET
  - 88 RID HOLE INDICATED ON OTHER THAN LAST RID
  - 89 DUPLICATE DISPLACEMENT FOR RID HOLE
  - 94 DUPLICATE DISPLACEMENT FOR RID CHAIN
  - 95 RID CHAIN COUNT NOT EQUAL TO RIDS IN CHAIN
  - 96 LAST RID CHAIN ENTRY DOES NOT HAVE ZERO NEXT
  - 97 CALCULATED DISPLACEMENT TO FREE SPACE DIFFERS
  - 98 CALCULATED FREE SPACE AFTER GARBAGE COLLECTION DIFFERS

Messages and explanations for integrity check errors

!DB/SMU issues the following messages for integrity check errors that it finds during a scan.

- **01 I/O ERROR FLAG ON IN HEADER**

  DB2 sets the X'80' flag in the first byte of the page if !DB/SMU previously encountered an I/O error when processing that page.

  If this flag is on, !DB/SMU displays message 01 together with the page header (the first 20 bytes of the page for table spaces or 28 bytes for index spaces).

- **02 PAGE END CHARACTER MISMATCH**

  When updating a page, DB2 switches the X'10' flag in both the first and last byte of the page. For a page to be valid, either the flag must be off in both places or it must be on in both places. This flag is used especially for 32K page sizes, since there are 8 VSAM CIs written for the one logical DB2 page. The first byte of the first page must match the last byte of the last page to ensure that all pages were written as a unit.

  If !DB/SMU indicates a mismatch, it displays message 02 together with the page header (the first 20 bytes of the page for table spaces or 28 bytes for index spaces).

- **03 PAGE NUMBER MISMATCH**

  DB2 records the number of each page in the page header. Pages are numbered relative to 0 since the first page is page 0. The page number is a 3-byte field at +X'08'. The page number in the data must match the physical location of the page.

  In a partitioned table space or the clustering index for a partitioned table space, the partition number is recorded in the high-order bits of this field; these bits must match, too.

  If there is a page number mismatch, !DB/SMU displays message 03 together with the page header (the first 20 bytes of the page for table spaces or 28 bytes for index spaces).

- **04 BROKEN PAGE FLAG IS ON**

  If DB2 detects a page integrity error, the broken page flag (X'80') is set at +X'0B' in the page header.

  !DB/SMU displays message 04 together with the page header (the first 20 bytes of the page for table spaces or 28 bytes for index spaces).

Messages and explanations for integrity check errors (continued)

- **05 FLAGS MISMATCH -- NOT AN INDEX**

  The page type flag must match the type of dataset being processed. If the flag (X'44' on at +X'0B') indicates that this is an index page, but the dataset is a table space, this mismatch signals an error.

  !DB/SMU displays message 05 together with the page header (the first 20 bytes of the page).

- **06 FLAGS MISMATCH -- INDEX FLAG NOT ON**

  The page type flag must match the type of dataset being processed. If the flag indicates that this is not an index page (X'44' is not on at +X'0B') but the dataset is an index space, the mismatch signals an error.

  !DB/SMU displays message 06 together with the page header (the first 28 bytes of the page).

- **07 FLAGS MISMATCH -- HEADER PAGE FLAG INCORRECTLY ON**

  Only the first page can be the header page. If the page type flag indicates that this is a header page (X'18' is on at +X'0B'), but it is not the first page of the dataset, a mismatch signals an error.

  !DB/SMU displays message 07 together with the page header (the first 20 bytes of the page for table spaces or 28 bytes for index spaces).

- **08 FLAGS MISMATCH -- HEADER PAGE FLAG NOT ON**

  The first page of the dataset must be the header page. If this is the first page of the dataset, but the flag does not mark this as a header page (X'18' is *not* on at +X'0B'.), a mismatch signals an error.

  !DB/SMU displays message 08 together with the page header (the first 20 bytes of the page for table spaces or 28 bytes for index spaces).

- **09 FLAGS MISMATCH -- SPACE MAP PAGE FLAG INCORRECTLY ON**

  Space map pages should occur at regular intervals; for 4K pages, the space map pages are at pages 1, 10760, 21520, and so forth. If the current page should not be a space map page, but the type flag marks this as a space map page (X'10' is on at +X'0B'), a mismatch signals an error.

  !DB/SMU displays message 09 together with the page header (the first 20 bytes of the page for table spaces or 28 bytes for index spaces).

Messages and explanations for integrity check errors (continued)

- **10 FLAGS MISMATCH -- SPACE MAP PAGE FLAG NOT ON**

  Space map pages should occur at regular intervals; for nonsegmented table spaces with 4K pages, the space map pages are at pages 1, 10760, 21520, and so forth. If the current page should be a space map page, but the type flag does not mark this as a space map page (for example, X'10' is not on at +X'0B'), this mismatch signals an error.

  !DB/SMU displays message 10 together with the page header (the first 20 bytes of the page for table spaces or 28 bytes for index spaces).

- **11 SPACE MAP COUNT INVALID**

  In space map pages, a header field specifies the number of dataset pages covered by this space map page. There are certain valid values for each page size (4K or 32K). If the value in this 2-byte field at +X'0C' (2 bytes for indexes and nonsegmented table spaces, 4 bytes otherwise) is not an allowable value, !DB/SMU signals an error.

  !DB/SMU displays message 11 together with the page header (the first 20 bytes of the page for table spaces or 28 bytes for index spaces).

- **12 FREESPACE VALUE INCORRECT, PAST MAP START**

  The free space cannot extend into the ID map area. The offset to the contiguous free space is a 2-byte field at +X'0E'.

  If the free space extends into the ID map area, !DB/SMU displays message 12 together with the page header (the first 20 bytes of the page).

- **13 NUMBER OF ID MAP ENTRIES = 0, OR > 127.**

  If the DB2 version number is less than Version 3.1, the maximum number of ID map entries is 127. If the DB2 version number is 3.1 or greater, than the maximum number of ID map entries is 255. The erroneous value is a 1-byte field at +X'12'.

  !DB/SMU displays message 13 together with the page header (the first 20 bytes of the page).

- **14 LARGE HOLE FOUND NOT ON HOLE CHAIN**

  A large hole (record code X'80') was found in the table space, but it is not on the chain of holes anchored from the prefix in a 2-byte field at +X'10'.

  There are two possibilities: either the record type flag could be damaged; or the hole chain anchor could be wrong.

  If error 16 (FREESPACE VALUE INCORRECT) is also issued for this page, the more likely error is that the record type flag is in error.

  !DB/SMU displays message 14 together with the six bytes at the hole record prefix.

Messages and explanations for integrity check errors (continued)

- **15 ERROR IN HOLE CHAIN**

  Holes in a page are chained together and anchored from a 2-byte field at +X'10' in the page. The chain is a 2-byte value at +X'03' in the hole and contains the offset within the page of the next large hole (code X'80'). The chain of large holes is incorrect if the offset from one hole to the next does not point to a large hole.

  If the chain is invalid, !DB/SMU issues message 15 together with the pointer to the incorrect area, for example, the anchor itself or the hole prefix in error.

- **16 FREESPACE VALUE NOT = TOTAL FREESPACE**

  The free space value in the header (a 2-byte field at +X'0C') must match the accumulated value of the free space in all the holes and the contiguous free space area.

  If the value is incorrect, !DB/SMU displays message 16 together with the page header (the first 20 bytes of the page).

- **17 INVALID RECORD TYPE**

  If the record code is not valid, !DB/SMU displays message 17. Valid record codes are as follows:

  X'00' Record
  X'04' Compressed row
  X'20' Overflow record
  X'40' Pointer
  X'80' Large hole

  Codes X'81', X'82', X'83', and X'84' represent small holes.

  !DB/SMU displays message 17 together with the record prefix (the first 6 bytes of the record).

- **18 RECORD LENGTH EXTENDS INTO FREESPACE**

  Large holes, records, and overflow records contain their length in a 2-byte value at +X'01' in the record prefix. !DB/SMU checks that the length is valid. If it is invalid, !DB/SMU displays error 17 (Invalid Record Type). If the record length extends past the start of the contiguous free space or into the ID map area of the page, !DB/SMU displays message 18.

  Together with the error message, !DB/SMU displays the record prefix (the first 6 bytes of the record).

Messages and explanations for integrity check errors (continued)

- **19 EMBEDDED ZERO PAGE**

  In response to a request from DB2, Media Manager formats all pages in one control area to binary zeros. DB2 uses these pages sequentially as rows are added to tables within the table space. All pages ever used by DB2 contain a header, a 2-byte trailer, at least one ID map entry, and at least one row or one hole. Even pages left empty because of FREEPAGE contain a header, a 2-byte trailer, and a free ID map entry.

  If a zero page is found embedded between pages used by DB2, !DB/SMU displays message 19, unless this is a segmented table space, where embedded zero pages are acceptable. !DB/SMU also displays the page header (the first 20 bytes of the page—all binary zeros).

- **20 DUPLICATE POINTER RECORD**

  If more than one pointer record points to the same overflow record, !DB/SMU displays message 20 for each record together with the pointer records in error.

- **21 POINTER REC DOES NOT POINT TO AN OVERFLOW RECORD**

  This error message arises from a cross-page check. If no overflow record was found to match a pointer record, !DB/SMU displays message 21 together with the pointer record.

  Since !DB/SMU analyzes spaces using VSAM CI processing independently of DB2, unmatched pointer records can occur if DB2 has retained committed updates in buffers; this is likely if the CLOSERULE for the space is N. If DB2 could not have had the space open during the !DB/SMU scan, these errors are real. Stop the table space and rerun the !DB/SMU scan. Errors that persist are real.

- **22 OVERFLOW REC ORPHANED -- NO MATCHING POINTER REC**

  This error results from a cross-page check. If an overflow record was found, but no pointer record exists for it, !DB/SMU displays message 22 together with the overflow record address.

  Since !DB/SMU analyzes spaces using VSAM CI processing independently of DB2, unmatched overflow records can occur if DB2 has retained committed updates in buffers; this especially likely if the CLOSERULE for the space is N. If DB2 could not have had the space open during the !DB/SMU scan, these errors are real. Stop the table space and rerun the !DB/SMU scan. Errors that persist are real.

Messages and explanations for integrity check errors (continued)

- **23 INVALID ID MAP ENTRY NUMBER**

  Records and overflow records contain a ID map entry number in a 1-byte value at +X'05' in their prefix. In turn, the ID map entry contains the offset to the start of the record. DB2 cannot find the record if the ID map entry value is not correct. If it is not, !DB/SMU displays message 23 together with the record prefix.

- **24 ID MAP FREE CHAIN INVALID -- MISSING FREE ENTRIES**

  If not all free ID map entries (either X'80' is on at +X'00' or the value is X'0000') are on the free chain anchored in the penultimate byte of the page, !DB/SMU displays message 24 together with the ID map area.

- **25 ORPHAN ID MAP ENTRY -- NO RECORD FOR ENTRY**

  Each ID map entry that is not free (X'80' is not on at +X'00' and the value is not X'0000') must point to a record. For each ID map entry that does not point to a record, !DB/SMU displays message 25 together with the ID map entry.

- **26 ID MAP FREE CHAIN INVALID -- FREE ENTRY NOT ON CHAIN**

  For each free ID map entry found (either X'80' is on at +X'00' or the value is X'0000') that is not in the free chain anchored in the penultimate byte of the page, !DB/SMU displays message 26 together with the ID map entry.

- **27 ID MAP FREE CHAIN INVALID -- POINTS TO NON-FREE ENTRY**

  If the free chain anchored in the penultimate byte of the page points to a ID map entry that is not flagged as free (X'80' is not on at +X'00' and the value is not X'0000'), !DB/SMU displays message 27 together with the ID map entry that points to the nonfree entry.

- **28 ID MAP FREE CHAIN INVALID -- LOOP IN CHAIN**

  If the free chain anchored in the penultimate byte of the page contains two entries that point to the same entry, thus creating a loop in the chain, !DB/SMU displays message 28 together with the second ID map entry that points to the doubly-addressed record.

- **29 KEYLENGTHS MISMATCH IN PAGE**

  DB2 keeps the keylength in two 2-byte fields at offsets +X'16' and +X'18' in the physical page header (the first 28 bytes of the page). The values must be the same. If they are not, !DB/SMU displays message 29 together with the physical page header (the first 28 bytes of the page).

Messages and explanations for integrity check errors (continued)

- **30 KEYLENGTH LARGER THAN 254**

  The DB2 keylength must not exceed 255 bytes in length. If it does, !DB/SMU displays message 30 together with the physical page header (the first 28 bytes of the page).

- **31 NO. SUBPAGES INVALID IN NON-SEGMENTED MODE**

  If the *segmented* flag is not on, the number of subpages should be one. If the number of subpages is not 1, !DB/SMU displays message 31 together with the physical page header (the first 28 bytes of the page).

  The *segmented* flag is the X'08' bit at offset +X'0C' in the page. The leaf page flag is the X'40' bit at +X'0C' in the page. The number of subpages is a 2-byte field at offset +X'1A' in the page.

- **32 NO. SUBPAGES HAS AN INVALID VALUE**

  In a segmented index page, the number of subpages must be 2, 4, 8, or 16. If it is not, !DB/SMU displays message 32 together with the physical page header (the first 28 bytes of the page).

  The number of subpages is in a 2-byte field at offset +X'1A' in the page.

- **33 AT LEAST ONE SUBPAGE IS NOT EMPTY IN EMPTY PAGE**

  If the *empty* flag is on in the physical page header, each subpage should be empty (that is, should contain no keys). If any is non-empty, !DB/SMU displays message 33 together with the physical page header (the first 28 bytes of the page).

  The *empty* flag is the X'10' bit at offset +X'0C' in the page.

- **34 NON-LEAF PAGES CANNOT BE SEGMENTED -- FLAG IS ON**

  Nonleaf pages cannot be segmented. If they are, !DB/SMU displays message 34 together with the physical page header (the first 28 bytes of the page).

  The *segmented* flag is the X'08' bit at offset +X'0C'in the page. The *leaf page* flag is the X'40' bit at offset+X'0C' in the page.

- **35 SUBPAGE IS MARKED BROKEN**

  DB2 maintains a bit in the logical page header if there were errors in the associated subpage. If this bit is on, !DB/SMU displays message 35 together with the logical page header. The broken subpage bit is the X'80' bit at offset +X'08' in the logical page header.

Messages and explanations for integrity check errors (continued)

- **36 KEY OFFSET VALUE IS INVALID**

  The logical subpage header contains the offset to the last (highest) key in the subpage. This offset value can be verified from another field in the logical page header—the number of keys in the subpage. If these two values are inconsistent, !DB/SMU displays message 36 together with the logical page header.

  The offset to the highest key in the subpage is a 2-byte field at offset +X'02' in the logical page header. Whereas the message indicates that the offset value is incorrect, it could be that the number of keys value is wrong. The number of keys is a 2-byte field at offset +X'00' in the logical page header.

- **37 FREESPACE OFFSET DOES NOT POINT TO FREESPACE**

  From the values for the number of keys and the offset to the last (highest) key, it is possible to verify the value for the offset to the free space in the subpage. If they are inconsistent, !DB/SMU displays message 37 together with the logical page header.

  The offset of the free space in the subpage is a 2-byte field at offset +X'06' in the logical page header. Whereas the message indicates that the offset value is incorrect, it could be that the number of keys value is wrong. The number of keys is a 2-byte field at offset +X'00' in the logical page header.

- **38 FREESPACE VALUE DOES NOT MATCH COMPUTED FREESPACE**

  Using the size of the subpage, it is possible to verify the value for the total amount of free space in the subpage against the offset to the free space. If they are inconsistent, !DB/SMU displays message 38 together with the logical page header.

  The amount of free space in the logical page is a 2-byte field at offset +X'04' in the logical page header, and the offset of the free space in the logical page header is a 2-byte field at offset +X'06' in the logical page header.

- **39 KEYS ARE NOT IN ASCENDING SEQUENCE**

  All the keys in any one subpage (or in a page if the index is nonsegmented) must be in ascending sequence. (Note that this is true even if descending order was specified for all or part of the key field when defined. DB2 inverts the key value for descending keys so that the values in the index are always ascending). Consecutive key entries can have the same key value only if the keys are non-unique.

  If the key sequence is not correct, !DB/SMU displays message 39 together with the key entry out of sequence.

Messages and explanations for integrity check errors (continued)

- **40 KEY ENTRY LENGTH ERROR -- NOT MATCH PREFIX DATA**

  In an index with non-unique keys, each key entry in a leaf page is prefixed with a 6-byte area that contains the length of the entry and the number of records indexed by this key. !DB/SMU checks the consistency of these two values using the known length of the key.

  If they are inconsistent, !DB/SMU displays message 40 together with the key entry.

- **41 SUBPAGES ARE NOT ALL THE SAME SIZE**

  In a segmented leaf page, each subpage must be the same size. The subpage directory contains the offset to the beginning of each subpage; the offsets must indicate subpages of equal size.

  If the subpages are not of equal size, !DB/SMU displays message 41 together with the subpage directory.

- **42 FIRST SUBPAGE DOES NOT FOLLOW SUBPAGE DIRECTORY**

  In a segmented leaf page, the first subpage must start immediately after the subpage directory. This is an additional check on the offsets in the subpage directory.

  If the offset is inconsistent with the page structure, !DB/SMU displays message 42 together with the subpage directory.

- **43 UNUSED INDEX PAGE REFERENCED BY KEY ENTRY**

  The space map pages in an index are used differently from those in a table space. In an index, each bit in the space map page indicates whether or not the corresponding index page is active.

  If one of the active key entries points to an inactive page, !DB/SMU displays message 43 together with the the entry from the referenced table. The problem could be with the pointer entry or with the space map bits. This error indicates that the current page is inactive, but has been referenced by a prior index pointer.

  This is a cross-page check. Since !DB/SMU analyzes spaces using VSAM CI processing independently of DB2, cross-page errors may be due to committed updates that DB2 has retained in buffers. This especially likely if the CLOSERULE for the space is N. If DB2 could not have had the space open during the !DB/SMU scan, these errors are real. Stop the space and rerun the !DB/SMU scan. Errors that persist are real.

Messages and explanations for integrity check errors (continued)

- **44 KEY ENTRY POINTS TO AN UNUSED INDEX PAGE**

  An inactive page is referenced by an index pointer. This error is detected in the page with the pointer record pointing to an earlier (lower-numbered) page that was determined to be inactive. !DB/SMU displays message 44 together with the page number of the page with a pointer to the unused page and the key entry (page number-key value).

  This is a cross-page check. Since !DB/SMU analyzes spaces using VSAM CI processing independently of DB2, cross-page errors may be due to committed updates that DB2 has retained in buffers. This is especially likely if the CLOSERULE for the space is N. If DB2 could not have had the space open during the !DB/SMU scan, these errors are real. Stop the space and rerun the !DB/SMU scan. Errors that persist are real.

- **45 INDEX PAGE DOUBLY REFERENCED**

  A page cannot be doubly-referenced. Two different index pointers cannot validly point to the same index page. If two do, one of these must be in error. !DB/SMU displays message 45 together with the page number of the second (higher-numbered) page with a pointer to the doubly-referenced index page and the reference table entry for the doubly-referenced index page, showing the page number of the first (lower-numbered) page that pointed to this entry.

  This is a cross-page check. Since !DB/SMU analyzes spaces using VSAM CI processing independently of DB2, cross-page errors may be due to committed updates that DB2 has retained in buffers. This especially likely if the CLOSERULE for the space is N. If DB2 could not have had the space open during the !DB/SMU scan, these errors are real. Stop the space and rerun the !DB/SMU scan. Errors that persist are real.

Messages and explanations for integrity check errors (continued)

- **46 PAGES PAST END OF INDEX ARE REFERENCED PREVIOUSLY**

  Pages past the allocated or used portion of the index cannot be validly referenced by another index page. If such a pointer is found, !DB/SMU displays message 46 together with the page number of the referenced page that exceeds the highest-used page and the table entry for the referenced page. The first word is the page number of the page that points to this page.

  This is a cross-page check. Since !DB/SMU analyzes spaces using VSAM CI processing independently of DB2, cross-page errors may be due to DB2 having extended the space while !DB/SMU was processing it. If DB2 could not have had the space open during the !DB/SMU scan, these errors are real. Stop the space and rerun the !DB/SMU scan. Errors that persist are real.

- **47 INDEX PAGE ORPHANED -- NOT REFERENCED BY ANY ENTRY**

  Some active page in the index is not pointed to by any other index page. If this condition exists, some chain has been broken. !DB/SMU displays message 47 together with the page number of the orphaned page and the table entry for the referenced page.

  This is a cross-page check. Since !DB/SMU analyzes spaces using VSAM CI processing independently of DB2, cross-page errors may be due to committed updates that DB2 has retained in buffers. This is especially likely if the CLOSERULE for the space is N. If DB2 could not have had the space open during the !DB/SMU scan, these errors are real. Stop the space and rerun the !DB/SMU scan. Errors that persist are real.

Messages and explanations for integrity check errors (continued)

- **48 NO. OF INDEX LEVELS DIFFERENT FROM PRIOR VALUE**

  DB2 uses a balanced index structure. Each index path from the root page to the leaf page should consist of the same number of levels. If a leaf page has been found that is not at the same level as all the prior leaf pages, !DB/SMU displays message 48 together with the page number of the leaf page and the table entry for the referenced page.

  This is a cross-page error. Since !DB/SMU analyzes spaces using VSAM CI processing independently of DB2, cross-page errors may be due to committed updates that DB2 has retained in buffers. This is especially likely if the CLOSERULE for the space is N. If DB2 could not have had the space open during the !DB/SMU scan, these errors are real. Stop the space and rerun the !DB/SMU scan. Errors that persist are real.

- **49 INDEX CHAIN DOES NOT GO DOWN TO LEAF PAGE LEVEL**

  If a page has been found whose index level cannot be determined, it follows that the page is not on any chain from leaf pages up to the root page. In these circumstances, some chain has been broken. !DB/SMU displays message 49 together with the page number of the page in error and the table entry for the referenced page.

  This is a cross-page error. Since !DB/SMU analyzes spaces using VSAM CI processing independently of DB2, cross-page errors may be due to committed updates that DB2 has retained in buffers. This is especially likely if the CLOSERULE for the space is N. If DB2 could not have had the space open during the !DB/SMU scan, these errors are real. Stop the space and rerun the !DB/SMU scan. Errors that persist are real.

- **50 MULTIPLE “FIRST” LEAF PAGES -- PRIOR POINTER = 0**

  There must be only one leaf page with a zero back chain. If there is more than one leaf page with a zero back chain, the message is issued for each one. If this error occurs, it is likely to occur at least twice.

  If more than one leaf page contains a zero back chain, !DB/SMU displays message 50 together with the page number of the leaf page with the zero back chain. For the duplicate *first* page, !DB/SMU displays the physical page header for the page. For the original *first* page, !DB/SMU displays only the page number.

Messages and explanations for integrity check errors (continued)

- **51 LOOP IN LEAF PAGE FORWARD CHAIN**

  A loop in the leaf chain is detected if more than one leaf page points to the same page. If this condition is found, !DB/SMU displays message 51 together with the page number of the second page that points to the same page. No additional data is displayed.

  Since !DB/SMU also checks leaf back chains, it must be the case that the first page was correct; otherwise !DB/SMU would display error message 53.

- **52 LEAF PAGE FORWARD CHAIN POINTS TO NON-LEAF PAGE**

  If a forward leaf chain word is pointing to a non-leaf page, the the chain is broken. !DB/SMU displays message 52 together with the page number of the page with the bad pointer. No additional data is displayed.

- **53 LEAF PAGE BACK CHAIN DOES NOT POINT TO PRIOR PAGE**

  The leaf back chain in the page pointed to must reference this leaf page. If it does not, the leaf chain is incorrect. !DB/SMU displays message 53 together with the page number of the page with the bad back leaf chain pointer. No additional data is displayed.

- **54 ORPHANED LEAF PAGE -- NOT ON FORWARD CHAIN**

  !DB/SMU checks for the completeness of the leaf chain. The chain is intact if each forward pointer points to a leaf page that points back to it. However, if some leaf pages are not on the chain, !DB/SMU displays message 54 together with the page number of the orphaned page. No additional data is displayed.

- **55 LOG RBA VALIDITY INDETERMINATE**

  Every time DB2 updates a page, the RBA of the log record reflecting the change is recorded in a 6-byte field at offset +X'01' in the page. The validity of this log RBA is itself reflected by the setting of a 1-byte field at +X'07' in the page. If the log RBA is valid, this field contains X'01'; if the log RBA is incorrect, this field contains X'FF'. If the value is neither of these, !DB/SMU displays message 55 together with the page header (the first 20 bytes of the page for table spaces or 28 bytes for index spaces).

  DB2 does not maintain this field reliably for header and space map pages, so !DB/SMU also accepts the value zero for these pages.

Messages and explanations for integrity check errors (continued)

- **56 SEGMENTED FLAG INCORRECTLY ON**

  The header page and space map pages from segmented table spaces are identified by the setting of the X'20' bit at offset +X'0B' in the page. If a table space is nonsegmented, but a page has this flag set, !DB/SMU displays message 56 together with the page header (the first 20 bytes of the page).

- **57 UPDATE STATUS INCONSISTENCY**

  When a page is modified, DB2 sets both the X'02' bit at +X'0B' in the page and the corresponding bit in the controlling space map page. The bits are reset only by an image copy, and should be set to the same value. If the bit in the page itself is on but the space map page bit is off, !DB/SMU displays message 57 together with the page header (the first 20 bytes of the page).

  The consequences of this error can be very serious. If the bit settings are inconsistent, the DB2 incremental image copy that uses the space map to identify modified pages will fail to copy this updated page.

  This is a cross-page check. Since !DB/SMU analyzes spaces using VSAM CI processing independently of DB2, mismatched update flags may be due to committed updates that DB2 has retained in buffers. If DB2 could not have had the space open during the !DB/SMU scan, these errors are real. Stop the table space and rerun the !DB/SMU scan. Errors that persist are real.

  If this error occurs during repair processing rather than scan processing, !DB/SMU may not accurately know the contents of the space map page. In order to rule this out, execute the CHK command against the prior space map page (page 1, 10760, and so forth), and CHK this page again.

  If this message occurs with any frequency, the simplest way to correct it is to run a DB2 full image copy.

Messages and explanations for integrity check errors (continued)

- **58 UPDATE STATUS INCONSISTENCY**

  When a page is modified, DB2 sets both the X'02' bit at +X'0B' in the page and the corresponding bit in the controlling space map page. The bits are reset only by an image copy and should be set to the same value. If the bit in the page itself is off, but the space map page bit is on, !DB/SMU displays message 58 together with the page header (the first 20 bytes of the page).

  As a result of this error, the DB2 incremental image copy that uses the space map to identify modified pages will copy this page again, unnecessarily.

  This is a cross-page check. Since !DB/SMU analyzes spaces using VSAM CI processing independently of DB2, mismatched update flags may be due to committed updates that DB2 has retained in buffers. If DB2 could not have had the space open during the !DB/SMU scan, these errors are real. Stop the table space and rerun the !DB/SMU scan. Errors that persist are real.

  If this error occurs during REPAIR processing rather than scan processing, !DB/SMU may not accurately know the contents of the space map page. Execute the CHK command against the prior space map page (page 1, 10760, and so forth), and CHK this page again.

  If this message occurs with any frequency, the simplest way to correct it is to run a DB2 full image copy.

- **59 INVALID OFFSET TO CONTIGUOUS FREE SPACE**

  The contiguous free space is the unused area between (1) the end of the rows and holes and (2) the beginning of the ID map area. It is anchored from a 2-byte field at offset +X'0E' in the page. If the value of this field exceeds the page size, !DB/SMU displays message 58. Unless there is no contiguous free space (that is, the ID map area begins immediately after the rows and holes), this message is usually accompanied by message 16.

  In addition to the error messages, !DB/SMU displays the page header (the first 20 bytes of the page).

Messages and explanations for integrity check errors (continued)

- **60 UNEXPECTED PAGE-TYPE FLAGS**

  The type of each page in every DB2 table space or index space is recorded in a 1-byte field at offset +X'0B' in the page. In a segmented table space, the header page must have the X'30' bits set. A space map page must have the X'38' bits set. Such pages may have only the X'82' bits set in addition. If the listed bit settings do not occur, !DB/SMU displays message 60 as well as the page header (the first 20 bytes of the page).

- **61 SEGMENT SIZE MISMATCH**

  The size of the segments in a segmented table space is recorded in a 2-byte field at +X'0E' in the space map page. This value must match the defined length of segments in the table space as recorded in SYSIBM.SYSTABLESPACE. If it does not, !DB/SMU displays message 61, the page header, and the space prefix (the first 24 bytes of the page).

- **62 INVALID SEGMENT SIZE**

  The size of the segments in a segmented table space must be a multiple of 4 pages; the minimum segment size is 4 pages, the maximum 64 pages. If the segment size recorded in the 2-byte field at +X'0E' in the space map page is invalid, !DB/SMU displays message 62, together with the page header and the space prefix (the first 24 bytes of the page).

- **63 INVALID COVERAGE**

  Within a segmented table space, for each combination of (1) page size (4K or 32K) and (2) segment size (such as 4*n* pages, where *n* is a number between 1 and 16 inclusive), a defined number of segments is *owned* by the space map page. If the number of segments, as recorded in the 2-byte field at +X'0C' in the space map page, is invalid, !DB/SMU displays message 63, together with the page header and the space prefix (the first 24 bytes of the page).

Messages and explanations for integrity check errors (continued)

- **64 SIZE - COVERAGE MISMATCH**

  The size of the segments in a segmented table space must be a multiple of 4 pages; the minimum segment size is 4 pages, the maximum 64 pages. The segment size is recorded in a 2-byte field at +X'0E' in the space map page. For each combination of (1) page size (4K or 32K) and (2) segment size, a defined number of segments is *owned* by the space map page, recorded in a 2-byte field at +X'0C' in the page. If both values are valid but do not correspond, !DB/SMU displays message 64, together with the page header and the space prefix (the first 24 bytes of the page).

- **65 INVALID FREE-SEGMENT COUNT**

  Of the segments *owned* by a space map page from a segmented table space, !DB/SMU calculates how many are free. If this differs from the value maintained by DB2 in a 2-byte field at +X'10' in the space map page, !DB/SMU displays message 65 together with the page header and the space prefix (the first 24 bytes of the page).

- **66 INVALID NEXT-SEGMENT**

  Of the segments in a segmented table space *owned* by a space map page, !DB/SMU determines which to allocate next. If this differs from the value maintained by DB2 in a 2-byte field at +X'12' in the space map page, !DB/SMU displays message 66 together with the page header and the space prefix (the first 24 bytes of the page).

- **67 INVALID LAST FORMATTED PAGE**

  Of those pages *owned* by a space map page in a segmented table space, !DB/SMU identifies the highest page that is or has ever been used. If it differs from the value maintained by DB2 (a 4-byte field at offset +X'14' in the space map page), !DB/SMU displays message 67 together with the page header and the space prefix (the first 24 bytes of the page).

- **68 UNDEFINED SEGMENT ENTRY FLAGS**

  The type of each segment in a segmented table space is recorded in a 1-byte field at +X'06' in the entry for the segment within the space map page. Only the X'C0' bits are defined. If any other bits are on, !DB/SMU displays message 68 and the segment entry prefix (the first 7 bytes of the segment entry).

Messages and explanations for integrity check errors (continued)

- **69 FREE SEGMENT ENTRY IS "FIRST"**

  The type of each segment in a segmented table space is recorded in a 1-byte field at +X'06' in the entry for the segment within the space map page. Free segment entries are denoted by the fact that the *in use* flag (X'80') is off. If a free segment entry also shows the X'40' flag, indicating that it is the first segment *owned* by the space map page for some OBID, !DB/SMU displays message 69 and the segment entry prefix (the first 7 bytes of the segment entry).

- **70 FREE SEGMENT ENTRY HAS "NEXT"**

  The type of each segment in a segmented table space is recorded in a 1-byte field at +X'06' in the entry for the segment within the space map page. Free segment entries are denoted by the fact that the *in use* flag (X'80') is off. Multiple segments *owned* by a space map page and allocated to the same OBID are chained together using a 4-byte field at offset +X'00' in the entry for the segment. If a free segment entry has a nonzero chain pointer, !DB/SMU displays message 70 together with the segment entry prefix (the first 7 bytes of the segment entry).

- **71 FREE SEGMENT ENTRY HAS OBID**

  The type of each segment in a segmented table space is recorded in a 1-byte field at +X'06' in the entry for the segment within the space map page. Free segment entries are denoted by the fact that the *in use* flag (X'80') is off. Used segments contain rows of only 1 table, whose OBID is stored in a 2-byte field at offset +X'04' in the entry for the segment. If a free segment entry has a nonzero OBID value, !DB/SMU displays message 71 together with the segment entry prefix (the first 7 bytes of the segment entry).

- **72 FREE SEGMENT ENTRY HAS SPACE INFO**

  The type of each segment in a segmented table space is recorded in a 1-byte field at +X'06' in the entry for the segment within the space map page. Free segment entries are denoted by the fact that the *in use* flag (X'80') is off. Each entry contains 4 bits for every page described by the entry. For example, the default segment size of 4 pages causes the segment entry to contain 2 bytes (four sets of 4 bits). If any of these bits is non-zero for a free entry, !DB/SMU displays message 72 together with the segment entry prefix (the first 7 bytes of the segment entry).

Messages and explanations for integrity check errors (continued)

- **73 ZERO OBID**

  The type of each segment in a segmented table space is recorded in a 1-byte field at +X'06' in the entry for the segment within the space map page. Used segment entries are denoted by the fact that the *in use* flag (X'80') is on. Used segments contain rows from only one table, whose OBID is stored in a 2-byte field at offset +X'04' in the entry for the segment. If a used segment entry has a zero OBID value, !DB/SMU displays message 73 together with the segment entry prefix (the first 7 bytes of the segment entry).

- **74 MULTIPLE "FIRST" SEGMENT ENTRIES**

  The type of each segment in a segmented table space is recorded in a 1-byte field at +X'06' in the entry for the segment within the space map page. Of the segments *owned* by the space map page, the first for each OBID shows the X'40' flag. If there are multiple *first* segments for an OBID, !DB/SMU displays message 74 together with the segment entry prefix (the first 7 bytes of the segment entry) for the second of these *first* segment entries.

- **75 SEGMENT ENTRY CHAIN IS BROKEN**

  If there are multiple segments owned by a space map page belonging to a segmented table space, and these segments are allocated to the same OBID, the segment entries are chained together using a 4-byte field at offset +X'00' in the entry for the segment within the space map page.

  If this chain contains an incorrect value, !DB/SMU displays message 75 together with the segment entry prefix (the first 7 bytes of the segment entry).

- **76 LOOP IN SEGMENT ENTRY CHAIN**

  If there are multiple segments owned by a space map page belonging to a segmented table space, and these segments are allocated to the same OBID, the segment entries are chained together using a 4-byte field at offset +X'00' in the entry for the segment within the space map page.

  If this chain contains a loop, !DB/SMU displays message 76 together with the segment entry prefix (the first 7 bytes of the segment entry) for the second of the two entries involved in the loop.

Messages and explanations for integrity check errors (continued)

- **77 FREE SEGMENT ENTRY ON CHAIN**

  If there are multiple segments owned by a space map page belonging to a segmented table space, and these segments are allocated to the same OBID, the segment entries are chained together using a 4-byte field at offset +X'00' in the entry for the segment within the space map page.

  If this chain points to a free entry, !DB/SMU displays message 77 together with the segment entry prefix (the first 7 bytes of the segment entry) for the entry that points to the free entry.

- **78 WRONG OBID ON CHAIN**

  If there are multiple segments owned by a space map page belonging to a segmented table space, and these segments are allocated to the same OBID, the segment entries are chained together using a 4-byte field at offset +X'00' in the entry for the segment within the space map page.

  If this chain points to an entry for a different OBID, !DB/SMU displays message 78 together with the segment entry prefix (the first 7 bytes of the segment entry) for the entry that points to the entry with the wrong OBID.

- **79 ORPHANED SEGMENT ENTRY**

  If there are multiple segments owned by a space map page belonging to a segmented table space, and these segments are allocated to the same OBID, the segment entries are chained together using a 4-byte field at offset +X'00' in the entry for the segment within the space map page.

  If this chain is broken, chained segment entries beyond the break cannot be located. In this case, !DB/SMU displays message 79 for each such segment entry together with the segment entry prefix (the first 7 bytes of the segment entry).

- **80 NON-ZERO SECTION**

  Within a space map page from a segmented table space, all bytes between the current end of the segment entries and the start of the update map should be zero. In addition, all bytes between the current end of the update map and a point 2 bytes before the page end should be zero. If either of these areas contains non-zero data, !DB/SMU displays message 80 together with the data at the start of the non-zero section.

Messages and explanations for integrity check errors (continued)

- **81 OBID INVALID FOR SEGMENT**

  All pages within one segment of a segmented table space contain rows from only one table, whose OBID is recorded in the corresponding segment entry in the preceding space map page. The OBID is the 2-byte value at +X'03'. For every row whose OBID does not match this value, !DB/SMU displays message 81 together with the 6-byte row prefix.

- **82 DATA PAGE BEYOND RANGE OF SPACE MAP**

  Each space map page in a segmented table space *owns* the following data pages. When the range of a space map page is exhausted, either another space map page should follow or there should be no more data. If instead there is a data page, !DB/SMU displays message 82 together with the page header (the first 20 bytes of the page).

- **83 SEGMENTED SPACE MAP PAGE FLAG INCORRECTLY ON**

  Space map pages from segmented table spaces are identified by the setting of the X'20' bit at offset +X'0B' in the page. If a table space is segmented and has this flag set in a non-space map page, !DB/SMU displays message 83 together with the page header (the first 20 bytes of the page).

- **84 CHAINED RID HAS RID HOLE FOLLOWS FLAG SET**

  If there is enough room to add a RID at the end of the list, DB2 sets a RID flag in the last RID of a nonunique index. RIDs in chains cannot have this flag set, since only one RID is allowed per chain entry. The data !DB/SMU displays is the RID.

- **88 RID HOLE INDICATED ON OTHER THAN LAST RID**

  If there is enough room to add a RID at the end of the list, DB2 sets a RID flag in the last RID of a nonunique index. Only the last RID can have this flag set, since all RIDs in a RID list must be contiguous. The data !DB/SMU displays is the RID.

- **89 DUPLICATE DISPLACEMENT FOR RID HOLE**

  !DB/SMU attempts to account for all space in an index page by building aq list of all accessible entries. In this case, !DB/SMU found that the location defined for a RID hole is already occupied by another page entity such as a key or chained free space. !DB/SMU displays the list entry that contains the displacement into the page followed by the length of the area. A RID hole is 5 bytes long.

Messages and explanations for integrity check errors (continued)

- **90 DBID MISMATCH**

  Every DB2 database has an internal database identifier (DBID), unique within the DB2 system. This value is recorded in a 2-byte field at offset +X'0C' in the header page for every table space and index space in the database. If the value does not match the database definition, !DB/SMU displays message 90. The most likely cause of this error is the incorrect use of a utility such as DSN1COPY to restore the space. In this case, !DB/SMU displays either message 91 (table spaces) or 92 (index spaces) with messae 90.

  The messages display with the page header (the first 20 bytes of the page for table spaces or 28 bytes for index spaces).

- **91 PSID MISMATCH**

  Every DB2 table space has an internal page space identifier (PSID), unique within the database. This value is recorded in a 2-byte field at offset +X'0E' in the header page for the table space. If the value does not match the table space definition, !DB/SMU displays message 91. The most likely cause of this error is the incorrect use of a utility such as DSN1COPY to restore the space. In this case, !DB/SMU may also display message 90.

  The messages display with the page header (the first 20 bytes of the page).

- **92 ISOBID MISMATCH**

  Every DB2 index space has an internal index space object identifier (ISOBID), unique within the database. This value is recorded in a 2-byte field at offset +X'0E' in the header page for the index space. If the value does not match the index space definition, !DB/SMU displays message 92. The most likely cause of this error is the incorrect use of a utility such as DSN1COPY to restore the space. In this case, !DB/SMU may also display message 90.

  The messages appear in the page header (the first 28 bytes of the page).

- **94 DUPLICATE DISPLACEMENT FOR RID CHAIN**

  !DB/SMU attempts to account for all space in an index page by building a list of all accessible entries. In this case, !DB/SMU found that the location defined for a RID chain entry is already occupied by a key, chained free space, or a RID hole. The data !DB/SMU displays is the list entry that contains the displacement into the page followed by the length of the area. A RID chain entry is 7 bytes long.

Messages and explanations for integrity check errors (continued)

- **95 RID CHAIN COUNT NOT EQUAL TO RIDS IN CHAIN**

  While following the RID chain, !DB/SMU found that the number of RIDs indicated in the chain header was not the same as the number of RIDs actually chained together. !DB/SMU displays the RID chain header. The first two bytes are the offset to the first chain entry. The second two bytes are the number of RIDs chained together.

- **96 LAST RID CHAIN ENTRY DOES NOT HAVE ZERO NEXT**

  A RID chain is present if the value for "number of RIDs following key" is negative or zero. The last entry in the RID list will be the RID chain header. The RID chain header contains the displacement to the first RID in the chain followed by the number of RID chain entries. Each RID chain entry consists of the 2-byte displacement of the next RID in the chain followed by the RID flags and the RID itself. If this error occurs, the last RID indicated by the RID count in the header did not have a zero next displacement, indicating that there is another RID in the chain beyond the number indicated in the header.

- **97 CALCULATED DISPLACEMENT TO FREE SPACE DIFFERS**

  !DB/SMU attempts to account for all space in the index page. The space left over should be contiguous free space pointed to at offset X'1C' in the page header. If this error occurs, !DB/SMU disagrees with DB2 as to where the contiguous free space begins. This is not a critical error but may result in DB2 not using space that is available within the page. The data !DB/SMU displays is the displacement it calculates.

- **98 CALCULATED DISPLACEMENT AFTER GARBAGE COLLECTION DIFFERS**

  !DB/SMU attempts to determine the amount of space that would be available after all RID chains, RID holes, and unusable space are eliminated, that is, if this single page were REORGed. The value for free space should match the value at X'1A' in the page header. This is not a critical error. DB2 uses this value to determine whether an internal REORG of the page will result in enough space to insert a new key or RID. The data !DB/SMU displays is the value it calculated.

- **99 ABEND PROCESSING PAGE**

  If !DB/SMU abends while processing a data page, it displays message 99 together with the page header (the first 20 bytes of the page). Processing continues. This message probably indicates that the page is severely damaged.

Appendix C. Extract Messages

Introduction

When you run a DB2 extract, !DB/SMU issues messages that update you on its progress. This appendix contains those messages.

Appendix Contents

Messages Issued by the Extract

Overview

These are the messages issued by the extract.

Messages

The system issues the following messages during the extract process.

EXTRACT STARTING

This message is written by the CLIST, and shows that the CLIST has been entered.

STARTING DB2 EXTRACT

This message is written after the DB2 thread has been initialized, and the extract program is about to start the dynamic SQL processing.

***nnnn* DATABASES EXTRACTED FROM SYSIBM.SYSDATABASE**

After all the databases have been extracted, this message shows the number of databases in the system.

***nnnn* TABLESPACE-PARTITIONS EXTRACTED FROM SYSIBM.SYSTABLESPACE**

After all the table spaces have been extracted, this message shows the number of table spaces in the system. One table space is counted for each partition of a partitioned table space.

***nnnn* TABLES EXTRACTED FROM SYSIBM.SYSTABLES**

After all the tables have been extracted, this message shows the number of tables in the system.

***nnnn* INDEXES EXTRACTED FROM SYSIBM.SYSINDEXES**

After all the index spaces have been extracted, this message shows the number of index spaces in the system.

***nnnn* KEYS EXTRACTED FROM SYSIBM.SYSKEYS**

After all the keys have been extracted, this message shows the number of keys in the system.

Messages (continued)

SQL ERROR CODE *-nn*
"sqlerrm"

This message signals that an error has been returned from DB2. Two messages are printed. The first prints the SQL error code, and the second prints the SQL error message returned from DB2. These messages are documented in the *IBM DB2 Messages and Codes* manual.

Appendix D. Data Sources

Introduction

This appendix contains information about the sources of data that !DB/SMU uses for its operations. It contains information about the system PDS, DB2 catalog data, dataset data, and extract messages.

Appendix Contents

Appendix Overview

Introduction

This appendix provides background information about the location of data that !DB/SMU uses in its operations.

Scope of information in this appendix

This appendix presents information on the system access to the following:

- System PDS use and members
- DB2 catalog data
- Extracts
- DB2 directory
- dataset data
- Extract mesages

!DB/SMU System PDS

Overview

!DB/SMU keeps its data in a PDS that is referred to as the !DB/SMU system PDS. This appendix decribes its functions and members. Refer to the Candle *!DB/Tools Installation and Customization Guide* for additional information about displaying and maintaining it.

Purpose of the system PDS

The !DB/SMU system PDS is created at installation time. It is used by !DB/SMU to keep all the data that is displayed on the panels. The names of the members in the PDS are usually unprintable values, which means that ISPF displays of the PDS are rather meaningless. In all other ways, however, it is a normal PDS: it can be copied by IEBCOPY or ISPF 3.3; it can be backed up by IEBCOPY; and it can be compressed with IEBCOPY or ISPF 3.1.

Caution

Do not run any compression routines against the system PDS unless you are planning to run a new extract.

Members in the system PDS

Members in the system PDS can contain

- Profiles
- Reports
- !DB/SMU options
- Extract history
- Extract data from the DB2 catalog

Viewing information about members of the system PDS

To view information about the members in the system PDS, take these steps:

Step	Action
1	On the !DB/SMU Primary Menu, select option H, Housekeeping.
2	Press Enter. **Result:** !DB/SMU displays the Administrative Functions panel.
3	On the Administrative Functions panel, select option 6, SYSPDS.
4	Press Enter. **Result:** !DB/SMU displays the Administrative Functions panel. This panel lists sets of related members according to database name.
5	To view information about the individual members of a given group, select that group by typing **S** or **R** next to its name.
6	Press Enter. **Result:** !DB/SMU displays information about the individual members of the group yo selected.

DB2 Catalog Data

Overview

The DB2 catalog is itself a DB2 database that contains the system tables that DB2 uses to manage all the user tables. The DB2 catalog also contains the definitions of the tables that make up the catalog. The DB2 catalog thus contains the information DB2 itself needs to process user SQL requests.

Extract data

On a schedule your site determines, !DB/SMU extracts data from the DB2 catalog using the program KTSDYNEX. Dynamic SQL is used to execute SQL queries that retrieve data which DB2 keeps about all the DB2 objects. (The DB2 objects !DB/SMU uses are databases, table spaces, tables, and index spaces.) The SQL queries are read from CNTL member EXTRCMDS. The results of the SQL queries are saved in the !DB/SMU system PDS member, DB2DATB. The previous DB2DATB member is renamed to DB2OLDD. Thus, the extract provides a point-in-time *snapshot* of your DB2 catalog.

!DB/SMU must use its own extract. It cannot use an extract that was generated for another product. See the *!DB/Tools !DB/Tools Installation and Customization Guide* for more information on performing batch catalog extracts.

When to run an extract?

It is necessary to run the extract only if table spaces, tables or indexes have been created or dropped.

How long does it take to run an extract?

Naturally, the time required for the !DB/SMU extract varies widely. If the extract can be scheduled in off hours, then the time will be both more predictable and much shorter.

DB2 Directory

Contents

The DB2 directory, database DSNDB01, contains the table spaces, tables, and indexes DB2 uses for internal processing.

Management

This database is not defined in the DB2 catalog, !DB/SMU does display DSNDB01 and its objects. All !DB/SMU functions are provided for the directory.

Dataset Data

Introduction

!DB/SMU displays comprehensive dataset data.

Space map pages

DB2 maintains free space and update information about the DB2 datasets themselves in *space map* pages. (A DB2 page is either 1 or 8 VSAM control intervals (CIs).) Space map pages occur at regular intervals throughout the DB2 dataset. The first is always the second page of the dataset. If the table space is nonsegmented with page size 4K, then there is another space map page every 10,760 pages. (Other intervals apply to table spaces with 32K pages and to segmented table spaces.) Index space space map pages occur about every 33,000 pages.

Table space space maps

Space map pages in nonsegmented table spaces contain 3 bits for each DB2 page in the table space. Two of these describe the amount of free space in the page, and the third indicates whether or not the page has been updated since the last image copy was taken for the table space.

The 2 free space bits per DB2 page can have 4 different values; the values are related to the size of the rows of the tables that are in the table space. The rows of each table have a certain computed size. DB2 uses the free space bits to indicate whether a page has room for the smallest of these rows, the average size row, the largest size row, or none of the rows in any of the tables in the table space. The 4 values of the 2 free space bits represent this information.

If a table space contains only one table, then the free space bits indicate only whether or not a page has room for the table row, that is, only 2 of the 4 possible values can appear.

Index space space maps

Index space space maps contain only 1 bit per DB2 page in the index space. This bit indicates whether or not the page contains index data or is empty. Index space space maps contain no free space information, and do not indicate whether a page has been updated since image copy. The bit indicates only whether the corresponding page is in use.

Reading the space maps

DB2 spaces are VSAM datasets, and can be processed with VSAM Control Interval processing. When a space is selected for a space map analysis, the VSAM dataset is opened. Dataset size and extent information is captured from the VSAM catalog. The space map pages in the DB2 space are read and the data copied to the !DB/SMU system PDS. This data is used to display the space map analysis information (space maps, space graphs, update maps, update graphs, usage maps, and usage graphs).

Scanning

When !DB/SMU is used to scan a DB2 table space or index space, VSAM CI processing is used. The VSAM dataset is opened; dataset size and extent information is captured from the VSAM catalog. The entire dataset is read up to the high-used RBA. Index spaces are read twice, since one pass is required to identify each page.

Since !DB/SMU analyzes spaces with VSAM CI processing independently of DB2, cross-page errors encountered during scan can be due to committed updates which DB2 has retained in buffers; this is especially likely if the CLOSERULE for the space is N. If DB2 could not have had the space open during the !DB/SMU scan, such errors are real. Otherwise, to determine whether this is the case, stop the space and rerun the !DB/SMU scan. Errors that persist are real.

Dataset security

In installations with an access control system such as RACF, you must be authorized to open and read the DB2 dataset. S913 abends occur if you are not authorized to open the dataset. !DB/SMU intercepts and recovers from these abends.

VSAM error messages

When the DB2 dataset is opened for !DB/SMU processing, the terminal may display VSAM error messages. These usually indicate that the dataset has not been properly closed by DB2. These messages can be ignored; they have no effect on the processing. Note that these messages recur each time the dataset is opened by !DB/SMU because the dataset is opened for read-only and not updated at CLOSE.

DASD extents display

!DB/SMU optionally displays the DASD extents for each DB2 dataset opened. The information displayed shows whether the DB2 dataset has been extended into multiple DASD extents, and whether those extents are scattered far away or close by.

This information is written to the file XTNTMAP, which can be allocated to the terminal by the command ALLOC FI(XTNTMAP) DA(*), either in the !DB/SMU CLIST itself (normally KTSXSMU) or by issuing it before the first scan, repair, or analysis.

Conversely, the display can be suppressed by removing the ALLOC statement. If the allocation is part of the !DB/SMU CLIST and you want to suppress the report, issue the following command from the command field before the first scan, repair or analysis: FREE FI(XTNTMAP).

Appendix E. Candle Customer Support

Introduction

Candle Corporation offers a comprehensive maintenance and support plan to ensure you realize the greatest value possible from your Candle software investments. We have more than 200 technicians worldwide, committed to providing you with prompt resolutions to your support requests.

Customer Support hours of operation are from 5:30 A.M. to 5:00 P.M., Pacific Time. In the event of an after-hours or weekend emergency, Candle's computerized call management system ensures that a technician will return your call within one hour. For customers located outside of North America, after-hours and weekend support is provided by Candle Customer Support locations in the United States.

Electronic Support

Candle provides information and support services using

- Candle's home page at **www.candle.com**. You can use the Candle Web site to
  - open problem records
  - access maintenance information
  - order products or maintenance
  - access IBM compatibility information
  - download fix packs for distributed products
  - read news and alerts
  - scan a list of scheduled Candle education classes
- Candle Electronic Customer Support (CECS), an electronic customer support facility. You can access this facility through the IBM Global Network. You can use CECS to
  - open problem records
  - search our database for solutions to known problems
  - look for answers to commonly asked questions
  - read news and alerts
  - scan a list of scheduled Candle education classes

Both CECS and the Candle Web site are available 24 hours a day, 7 days per week.

Telephone Support

Our support network consists of product specialists who work with you to solve your problem.

Candle uses an online problem management system to log and track all support requests. Your request is immediately routed to the appropriate technical resource.

When you call to report a problem, please have the following information:

- your Candle personal ID (PID) number
- the release level of the Candle product
- the release level of IBM or other vendor software
- identifying information and dates of recently applied maintenance to your Candle product or IBM product
- a detailed description of the problem (including the error message) and the events preceding the problem
- a description of any unusual events that occurred before the problem

Customer Support Phone Numbers

	Telephone	Fax
North America	(800) 328-1811	
	(310) 535-3636	(310) 727-4204
Europe		
Belgium/Luxembourg	+32 (0) 3 270 95 60	+32 (0) 3 270 95 41
France	+33 (0) 1 53 61 60 60	+33 (0) 1 53 61 06 16
Germany/Switzerland/ Austria	+49 (0) 89 54 554 333	+49 (0) 89 54 554 170
Italy - Freephone	800 780992	
Netherlands	+31 (0) 30 600 35 50	+31 (0) 30 600 35 10
Scandinavia	+46 (0)8 444 5940	+46 (0)8 623 1855
U.K.	+44 (0)161 437 5224	+44 (0)161 437 5225
(Southern Europe, Middle East and South Africa Agents call U.K.)		
Asia Pacific - English Hub		+61 2 9954 1818
Australia	+61 2 8912 9898	
Hong Kong	800 908 457	
India	+61 2 8912 9898	
Indonesia	0018 03061 2061	
Malaysia	1800 803 459	
New Zealand	0800 449 596	
Philippines	1800 1612 0096	
Singapore	800 616 2075	
Thailand	0018 00612 1045	
Asia Pacific - Japanese Hub	+81 3 3595 7150	+81 3 3595 7110
Asia Pacific - Korean Hub	+82 2 552 8744	+82 2 552 8746
Asia Pacific - Mandarin Hub	+88 62 2739 3223	+88 62 2378 5993
Asia Pacific e-mail address: ap_support@candle.com		

When your local support office is unavailable, you can contact Candle's North America support center. If USADirect® service is available in your country, use the 800 telephone number. If USADirect service is not available, ask your international operator for assistance in calling Candle's local (310) number.

Incident Documentation

You may be asked to send incident documentation to the Candle Customer Support Center. On the outside of all packages you send, please write the incident number given to you by the Customer Support representative.

Send tapes containing the incident information to the following address, unless directed otherwise by your Customer Support representative:

Candle Customer Support
Candle Support Center, *Incident number*
201 North Douglas Street
El Segundo, CA 90245

Send all other relevant documentation, such as diskettes or paper documentation, to the address provided by your Customer Support representative.

Ensuring Your Satisfaction with Customer Support

Candle Customer Support is committed to achieving high customer satisfaction ratings in all areas. These include

- connecting you to a support representative promptly
- providing you with the appropriate fixes
- answering support questions
- filling your shipping orders
- supplying documentation

If you have a concern that has not been resolved to your satisfaction, you can open a complaint ticket. All tickets are logged and tracked to ensure responsiveness and closure. Using the ticket information, a manager will contact you promptly to resolve your problem.

Special Characters

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

X

Z

User Comment Form

!DB®/SMU for DB2 User's Guide Version 500

TS54-5846-3

Please take a moment to share your comments and suggestions regarding Candle's documentation. Be as specific as possible.

	Page #	Comment
①		
②		
③		
④		
⑤		
⑥		
⑦		

Please provide the following information. Thank you for your time and assistance.

Name

Job Function

Company Site ID

Address

You may return your comments to us at your convenience. Upon receipt, we will respond to you within five business days.

FAX
(310) 727-4131

Fold, tape, and mail
(postage paid)
Please do not staple.

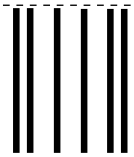
May we call to discuss your comments?

☐ YES Phone number: _____ Best time to call: _____

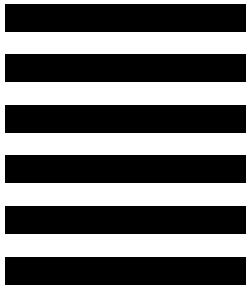
☐ NO

Cut or fold along line

Fold and tape *Please do not staple* Fold and tape



NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES



BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO 559 SANTA MONICA CA

POSTAGE WILL BE PAID BY ADDRESSEE

Candle®

ATTN INFORMATION DEVELOPMENT
CANDLE CORPORATION
201 N DOUGLAS STREET
EL SEGUNDO CA 90245

Fold and tape *Please do not staple* Fold and tape

Cut or fold along line


TS54-5846-3