



SERENA[®] **COMPAREX[®] 8.7 for z/OS**

Getting Started Guide

Serena Proprietary and Confidential Information

Copyright

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

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

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

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

Trademarks

Serena, TeamTrack, StarTool, PVCS, Collage, Comparex, Dimensions, Serena Dimensions, Mashup Composer, Mashup Exchange, Prototype Composer, Mariner and ChangeMan are registered trademarks of Serena Software, Inc. The Serena logo, Version Manager, Meritage and Mover are trademarks of Serena Software, Inc. All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

U.S. Government Rights

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

Publication date: April 2011

Table of Contents

Welcome to the Serena Comparex Getting Started Guide . . .	5
Guide to Comparex Documentation	5
Comparex Documentation Suite	5
Accessing the Documentation	6
Using PDF Manuals	6
Full-Text Search in Multiple Manuals	6
Typographical Conventions	7
Code Conventions	7
Introduction	9
Comparex Introduction	9
Comparex Overview	9
Types of Files You Can Compare	10
How Comparex Processes Information	11
Running Comparex: All-Defaults or Keywords	12
All-Defaults	12
Keywords	12
DATA or TEXT File Comparison	12
DATA	12
TEXT	13
Which to Use: DATA or TEXT	13
Environment Definitions	13
Getting Started	15
How Comparex Processes Information	15
INPUT	16
OUTPUT	16
Running Comparex: All-Defaults or Keyword	16
DATA or TEXT File Comparison	17
How DATA and TEXT Differ	17
Which to Use: DATA or TEXT	18
Getting Started: Create Two Known Files	18
DATA File Comparisons	19
DATA Comparison Summary	19
DATA Comparison Scenario: Mostly Defaults	19
Example Job Card Request and Output	19
Interpreting a DATA Difference Report	21
What If the Comparison Job Fails?	22
DATA Comparison Scenario: KEY Synchronization	23
Example Job Card Request and Output	23

Interpreting a DATA Difference Report with KEY, FORMAT, and MASK Keywords	25
TEXT Files	27
TEXT Comparison Summary	27
TEXT Comparison Scenario	27
Job Card Syntax for a TEXT Compare Request	27
Example TEXT Difference Report	28
Interpreting a TEXT Difference Report	29
Interfaces	31
Alternative Comparex Interfaces	31
Generating Delta Decks	31
ISPF Interface	32
Connected to Main Menu	33
Index	37

Welcome to the Serena Comparex Getting Started Guide

Audience and scope This book describes how to begin using Comparex and guides you through some commonly used comparisons. It contrasts DATA and TEXT comparison logic and also describes the components of difference reports.

This book also shows you how to:

- Run a DATA comparison with keys
- Run a DATA comparison without keys
- Run a TEXT comparison
- Understand the printed output of each comparison type

Navigating this book This book is organized as follows:

- Chapter 1 introduces and describes the general flow of Comparex.
- Chapter 2 provides information for you to begin using Comparex.
- Chapter 3 describes the use of the DATA keyword for comparing data.
- Chapter 4 describes the use of the TEXT keyword for text comparisons.
- Chapter 5 discusses the various interfaces for Comparex.

Guide to Comparex Documentation

The following sections provide basic information about Comparex documentation.

Comparex Documentation Suite

The complete Comparex documentation suite for z/OS installations consists of the following manuals:

Manual	Description
<i>Serena[®] Comparex[®] for z/OS Installation Guide</i>	Installation requirements and instructions for installing Comparex under z/OS.
<i>Serena[®] Comparex[®] for z/OS Getting Started Guide</i>	Quick-start guide to basic Comparex functions.
<i>Serena[®] Comparex[®] for z/OS User's Guide</i>	Comprehensive guide to using Comparex functions in batch and interactive environments under z/OS.
<i>Serena[®] Comparex[®] Quick Reference</i>	Cross-platform summary booklet with command formats and keyword descriptions.

Accessing the Documentation

All Comparex documentation is provided electronically in Adobe Portable Document Format (PDF) format. Licensed customers may download the documentation from the Serena Customer Support web site at www.serena.com/support.

Copies of the *Installation Guide* for each of the supported operating system families are also provided on the physical product media in the Documentation folder. A brief Readme file in this directory provides links to the full set of online documentation and to the latest comprehensive Readme file for the product. The Readme is frequently updated after the initial product release and contains the latest product bulletins and updates.

Printed copies of the *Installation Guide's* and the *Quick Reference* booklet are shipped in the box with the physical product media. Additional copies may be purchased separately on request to Customer Support.

Using PDF Manuals

The Comparex manuals are provided as Adobe Portable Document Format (PDF) files. To view PDF files, use Adobe® Reader®, which is freely available at www.adobe.com.



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

The PDF manuals include the following features:

- **Bookmarks.** All of the manuals provide bookmarks that work like a hypertext table of contents, so you can jump quickly to specific topics. By default, the bookmarks are displayed in a separate pane to the left of the manual pages.
- **Links.** Cross-reference links in the body of the books enable you to jump to other, related topics with a single mouse click. These links are highlighted in blue.
- **Printing.** While viewing a manual, you can print the current page, a range of pages, or the entire manual.
- **Multi-document search.** Adobe Reader 6 and later includes an advanced search feature that enables full-text search for the same word or phrase across multiple PDF files at once.

Full-Text Search in Multiple Manuals

To search for a word or phrase anywhere in the text of several PDF documents at once, perform the following steps:

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



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

Typographical Conventions




The following typographical conventions are used in the online manuals. These typographical conventions are used to assist you when using the documentation; they are not meant to contradict or change any standard use of typographical conventions in the various product components or the host operating system.

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

Code Conventions

This legend describes the symbols and abbreviations used in the descriptions of the Comparex keywords. The symbols and abbreviations are used the same way in the *Comparex Quick Reference* and in the *Comparex Getting Started Guide*.

Symbol	Meaning
[]	Brackets enclose an optional entry.
()	Parentheses must be coded as shown in the examples.
{ }	Braces indicate a required entry when more than one selection is available.

Symbol	Meaning
CAPS	Uppercase letters indicate a keyword, name, or field to be coded as shown.
lowercase	Lowercase letters indicate that variable information is to be supplied.
underscore	Underscores indicate the default value.
ddd	Relative displacement from the first position of the input record. When MODE=SYSTEM, displacements are relative to 0, and values range from 0 to 32767. When MODE=APPLICATION, displacements are relative to 1, and values range from 1 to 32767.
len	Length, in bytes. Values range from 1 to 32767. (For KEY and KEY1, values range from 1 to 256.)
t	Type. Values are X for hexadecimal and C for character.
vvvv	Literal value between apostrophes. For example, t'vv' could be X'5B'.
N=	Descriptive phrase for display on Comparex report. Maximum length is 32 bytes.
	Applies to a DATA comparison.
	Applies to a TEXT comparison.
	Applies to a DIRECTORY comparison.




Chapter 1

Introduction

Comparex Introduction	9
Comparex Overview	9
Types of Files You Can Compare	10
How Comparex Processes Information	11
Running Comparex: All-Defaults or Keywords	12
DATA or TEXT File Comparison	12
Environment Definitions	13

Comparex Introduction

Comparex is a utility designed to compare two files and print a report showing the records that are different. It helps you check the accuracy of maintenance changes before implementation and it facilitates effective unit, system, and regression testing for new development. This utility performs specific field comparisons, highlights the differences, and generates a report that underscores those differences. You can perform DATA, TEXT, and DIRECTORY compares.

With This Kind of Compare...	You Can Compare...
DATA 	Virtually all databases, CSECTs, master/transaction files, load modules
TEXT 	Source code, JCL, reports, documentation
DIRECTORY 	Library Management systems: directories



NOTE DATA comparison is the default.

Data processing personnel in charge of the accuracy of systems implementations and modifications can use Comparex to help ensure that their work is correct and complete.

Comparex Overview

When a change is introduced into a system, Comparex compares files produced before the change with files produced after the change. The difference report is printed, allowing you

to see precisely which bytes differ between the two files, record by record. The results will confirm your expected changes and, more importantly, expose any unexpected changes.

An effective Comparex job can be run in all-defaults mode, with no keywords. Comparex reads two files, comparing a record number from the first file with the same record number from a second file. It then prints both records if it finds one or more differing bytes. In all-defaults mode, Comparex reads to the end of each file, printing all the pairs of differing records and all the extra records from the longer input file.

The most frequently requested options are implemented as the Comparex defaults. The all-defaults mode can be significantly modified. Using extensive optional keywords, you can modify input processing, record pairing, and printing routines in Comparex. In a keywords job, you specify free-form keywords to change the default parameters, so a file of any organization can be read, synchronization can be done by logical keys, and the format of the report can be customized. In addition, keywords can be used to tell Comparex to create an output file of selected records.

Types of Files You Can Compare

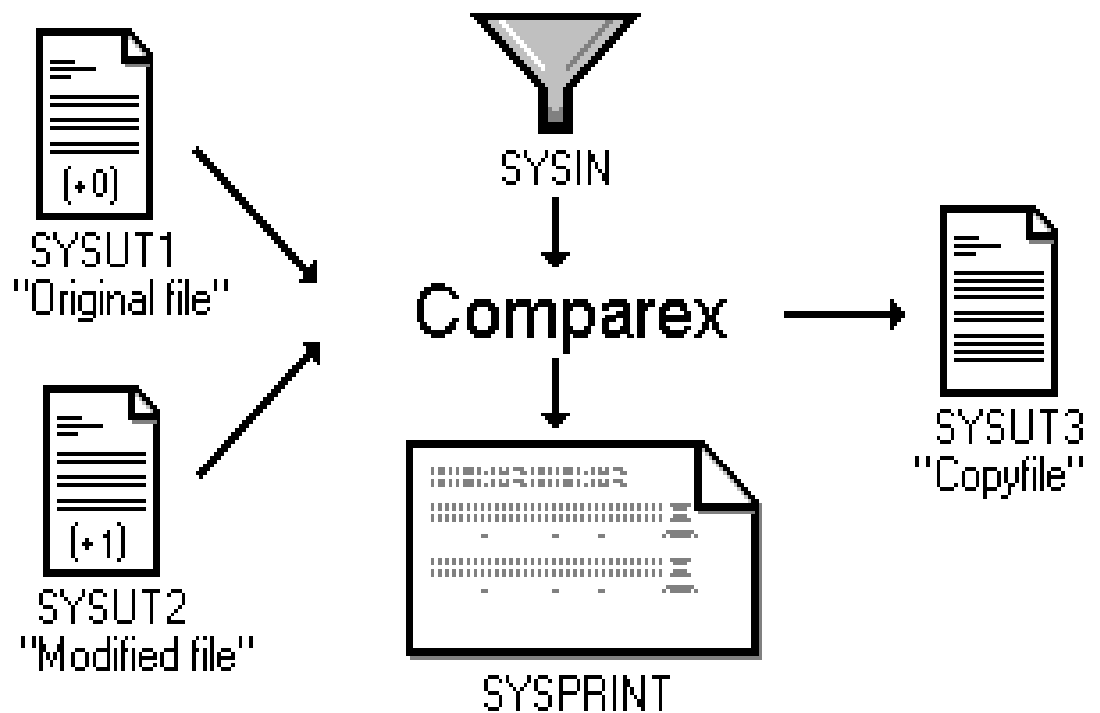
Comparex can compare any two files of almost any structure or organization. Comparex can directly process the following types of files:

- Program
 - Source Code
 - Object Code
 - Executable Load Modules
 - CSECT Parsing or Undefined Block to Block
- Job Control Language (JCL)
- CLIST
- System Master Files
 - Sequential
 - QSAM
 - ISAM
 - VSAM
- System Intermediate Files
- System Transaction Files
- Directories, Selected Members, Ranges of Members, or All Members
 - CA-Panvalet
 - CA-Librarian
 - FUJITSU/FACOM's GEM
 - Partitioned Data Set (PDS/PDSE)
 - OWL

- CA-ROSCOE
- WYLBUR
- IMS, IMS/FASTPATH
- CICS
- DB2
- FOCUS
- ORACLE
- Control Card Images
- Reports
- Documentation

How Comparex Processes Information

The following diagram illustrates the basic flow of the Comparex utility.



Running Comparex: All-Defaults or Keywords

All-Defaults

Comparex can run in “all-defaults” mode. In this mode, Comparex reads two files, comparing a record from the first file with the same number record from a second file. It then prints both records if it finds one or more differing bytes.



NOTE In this mode, Comparex compares records from a DATA file without attempting to match on keys.

Keywords

You can modify any of the defaults with Comparex keywords. Using these keywords, you can tell Comparex to read a file of any organization, synchronize by logical keys, choose records and fields to compare, customize the report format, and create an output file of selected records.



NOTE The examples that follow this chapter are very simple, using a minimum number of keywords. We recommend that you experiment with these samples before customizing Comparex with the wide range of keywords available.

DATA or TEXT File Comparison

Comparex recognizes two file categories: DATA and TEXT. Each category uses different programming logic because the type of synchronization differs. If neither category is specified, Comparex defaults to DATA.

DATA

To Comparex, a DATA file contains formatted records. DATA files have fields in fixed positions within each record. DATA records are generally sequenced (usually defined by a key field). Comparex compares DATA files by pairing physical records and then comparing these records field by field, using key-, segment-, or record-number to record-number synchronization.

Examples of DATA files are:

- Master files
- Intermediate files
- Transaction files
- Databases
- Load modules

TEXT

To Comparex, TEXT is any file where no specific format exists within the record. TEXT files do not have bytes and fields in fixed positions. TEXT records can contain blanks and might be entirely free-form. Rather than pairing records by record number or key, TEXT synchronization matches records by content to find blocks of matching text and isolate differences. Examples of TEXT files are:

- Program source code
- JCL
- Reports
- Documentation

Which to Use: DATA or TEXT

Run a DATA comparison under the following situations:

- If files have keys, use DATA with KEY synchronization
- If files are databases, use DATA with KEY or SEGMENT synchronization



NOTE Avoid Random KEYs or segments, if possible, as this is extremely inefficient.

- If files are object code, use DATA without a key
- If data sets are load modules, use the CSECT Parsing option

Use TEXT if none of these situations apply. Maximize the buffering area required for look-ahead logic and minimize the number of records that must match "exactly," until you obtain the desired results. See the *Comparex for z/OS User's Guide* chapter on "Decisions about DATA and TEXT" for more information.



NOTE If TEXT has neither been specified nor been nullified because of inconsistencies, Comparex uses DATA comparison logic. When Comparex performs DATA comparison logic, it cannot also perform TEXT comparison logic in the same run. The two logic routines are mutually exclusive.

Environment Definitions

Comparex runs without modification under the following operating systems:

- | | | |
|-----------|----------|-----------------------------|
| ■ z/OS | ■ z/VM | ■ z/VSE |
| ■ OS/390 | ■ VM/ESA | ■ VSE/ESA |
| ■ MVS/ESA | ■ VM/SP | ■ VSE/SP |
| ■ MVS/XA | ■ VM/XA | ■ VSE/XA |
| | | ■ DOS/VSE |
| | | ■ Software Pursuits MVT/VSE |
| | | ■ Nixdorf NIDOS |

Chapter 2

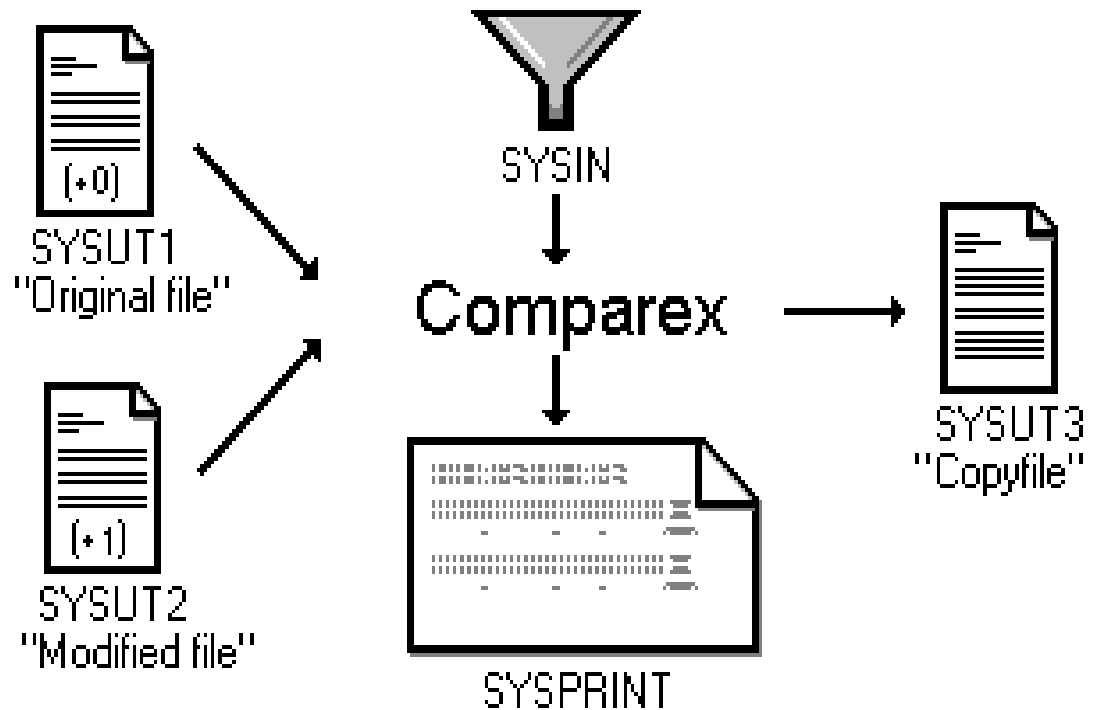
Getting Started

How Comparex Processes Information	15
Running Comparex: All-Defaults or Keyword	16
DATA or TEXT File Comparison	17
Getting Started: Create Two Known Files	18

How Comparex Processes Information

The following figure illustrates the basic flow of the Comparex utility.

Figure 2-1.
Comparex
General
Flowchart



As [Figure](#) illustrates, Comparex first reads the input files (SYSUT1 and SYSUT2) and compares each record from the first file with the corresponding record on the second file. When it detects at least one differing record, Comparex prints a *difference report*. This report displays both input records and underscores the differences.

The section below provides more details on the input and output files.

INPUT

Comparex uses the following three files:

- SYSUT1** SYSUT1 is the first file to be compared. Generally, this is the original file or old master. If you are comparing program source code, SYSUT1 is the unmodified version.
- SYSUT2** SYSUT2 is the second file to be compared. Generally, this is the modified file or new master. If you are comparing program source code, SYSUT2 is the updated version.
- NOTE** If you plan to process CA-Panvalet, CA-Librarian, GEM, or OTHER file structures directly, be sure to supply the proper ddname, instead of SYSUT1, SYSUT2, or both. This enables the Comparex interface (CPXIFACE) to access the file structure.
- SYSIN** As delivered, Comparex operates in an 'all-defaults' mode (see ["Running Comparex: All-Defaults or Keyword" on page 16](#)). The most frequently used keywords are already specified. With the optional SYSIN file, Comparex enables you to override the default keyword values:
- If SYSIN is either empty or cannot be opened, Comparex operates with the default values.
 - If SYSIN is not empty, Comparex modifies its default processing with the specified keyword values.

OUTPUT

Comparex generates the following two files:

- SYSPRINT** This output file contains the difference report, which lists the comparison results, the keyword defaults, and the end-of-job statistics.
- SYSUT3** Optionally, you can request a visual representation of each record type identified. Moreover, you can generate a 'selected records' file, SYSUT3. This file contains records that are missing from SYSUT2 and functions as a delta deck to update the original file. The records contained in SYSUT3 include the following:
- Matched records where differences in data are found
 - Records that are inserted on SYSUT2. (Records inserted on SYSUT1 do *not* go to SYSUT3.)
- You can also specify up to 5 different output files in place of SYSUT3. See the COPYSPPLIT keyword in the *Comparex User's Guide* for more information.

Running Comparex: All-Defaults or Keyword

Comparex can run in an "all-defaults" mode. In this mode, Comparex reads two files, comparing each record from the first file with the same numbered record on the second file. Comparex then prints both records only when it finds at least one byte that differs. Be

aware that in this mode, Comparex compares records within a DATA file without any attempt to match on keys.

You can modify the all-defaults mode with Comparex keywords. In a keywords job, you specify free-form keywords to change the default parameters. Through keywords, you can invoke Comparex to read a file of any organization, to synchronize by logical keys, to customize the report format, and to create an output file of selected records.

DATA or TEXT File Comparison

Comparex recognizes two file categories: DATA and TEXT. Each category requires a different logic routine, because the type of synchronization differs. If neither category is specified, Comparex assumes the default (DATA).



NOTE In an “all-defaults” job, Comparex performs DATA comparison logic. Comparex does *not* perform TEXT file comparison logic.

How DATA and TEXT Differ

DATA

Comparex defines DATA as any file that contains a known inter-record relationship. These files have bytes and fields in fixed positions on each record. Comparex compares these files record to record, using key, segment, or physical-record to physical-record synchronization.

Examples of DATA files are:

- Master files
- Intermediate files
- Transaction files
- Databases
- Load modules

TEXT

Comparex defines TEXT as any file that does not contain inter-record relationships. TEXT files do not have bytes and fields in fixed positions. Rather than pairing records for comparison, TEXT synchronization uses buffers and look-ahead logic to pair blocks and isolate differences.

Examples of TEXT files are:

- Program source code
- JCL
- Reports
- Documentation

Which to Use: DATA or TEXT

You should run a DATA comparison under the following situations:

- If files have keys, use DATA with KEY synchronization.
- If files are databases, use DATA with KEY or SEGMENT synchronization.



NOTE Avoid Random KEYS or segments if you can, as this is extremely inefficient.

- If files are object code, use DATA without any key.
- If data sets are load modules, use the following CSECT Parsing option to DATA:
DATA=CSECT or DATA=(CSECT,ADCON=NO) /* See User's Guide

If none of the situations apply, you should use TEXT, maximizing the buffering area required for look-ahead logic and minimizing the number of records that must match "exactly," until you get the desired results. See the *Comparex z/OS User's Guide*, Chapter 4, "Decisions about DATA and TEXT" for more information.



NOTE If TEXT has either not been specified or has been nullified because of inconsistencies, Comparex uses DATA comparison logic. Furthermore, when Comparex performs DATA comparison logic, it cannot also perform TEXT comparison logic in the same run; the two logic routines are incompatible.

Getting Started: Create Two Known Files

To perform the basic scenarios in the subsequent chapters, allocate and create the two data sets shown below.

somnode.FILE1

```
RECORD01AAAAAAAAAAAAA
RECORD02BBBBBBBBBBBBBB
RECORD03CCCCCCCCCCCCC
RECORD04DDDDDDDDDDDDDD
RECORD05EEEEEEEEEEEEEE
RECORD06FFFFFFFFFFFFFFF
RECORD07GGGGGGGGGGGGGG
RECORD08HHHHHHHHHHHHHH
RECORD09IIIIIIIIIIIIII
RECORD10JJJJJJJJJJJJJJ
RECORD26 INSERTION
```

somnode.FILE2

```
RECORD01AAAAAAAAAAAAA
RECORD02BBBBBBBBBBBBBB
RECORD03FIRST CHANGE
RECORD04DDDDDDDDDDDDDD
RECORD05EEEEEEEEEEEEEE
RECORD06SECONDCHANGE
RECORD07GGGGGGGGGGGGGG
RECORD08HHHHHHHHHHHHHH
RECORD09IIIIIIIIIIIIII
RECORDXX INSERTION
RECORD10JJJJJJJJJJJJJJ
RECORD11 INSERTION
```

Figure 2-2. Sample Files - FILE1 And FILE2

Chapter 3

DATA File Comparisons

DATA Comparison Summary	19
DATA Comparison Scenario: Mostly Defaults	19
DATA Comparison Scenario: KEY Synchronization	23

DATA Comparison Summary

To Comparex, DATA is any file containing formatted records. DATA files have bytes and fields in fixed relationships in their records and often include some type of key field for indexing purposes. Comparex compares DATA files by pairing physical records, aligning records by offset or field, then comparing content on a byte-to-byte basis.

Comparex accepts keyword parameters that designate comparison options. The examples in this chapter are very simple, using a minimum number of keywords. We recommend that you experiment with these samples before customizing Comparex with the wide range of keywords supported by this product. Refer to the *Comparex User's Guide* for details on about keyword parameters.

DATA Comparison Scenario: Mostly Defaults

In this scenario, we will run Comparex in a mostly default mode to compare the two files you created in ["Getting Started: Create Two Known Files"](#) on page 18. In this mode, Comparex reads two files comparing each record from the first file with the same numbered record on the second file. Comparex then prints both records *only* when it finds at least one byte that differs.

Example Job Card Request and Output

The following job card ([Figure 3-1](#)) requests a simple DATA comparison that assumes default values for most keywords. By default, Comparex does not attempt to align record formats or match on keys. Once Comparex finds an out-of-sync situation, it prints all remaining records as differing, so to avoid a large printout, we specify MAXDIFF=5.

Figure 3-1.
Simple DATA
Compare Request

```
//MVS$JOB JOB (accounting),COMPAREX,CLASS=A
//CPX EXEC PGM=COMPAREX
//SYSPRINT DD SYSOUT=*
/*
//SYSUT1 DD DISP=SHR,DSN=somnode.FILE1
//SYSUT2 DD DISP=SHR,DSN=somnode.FILE2
//SYSIN DD *
MAXDIFF=5 /* to avoid a runaway compare */
CONTINUE /* to see final count of differences */
/*
```


Interpreting a DATA Difference Report

This section discusses the main features of the difference report. The following discussion applies to both the default mode and keyword mode of operation.

Keywords

On the first page of the report, in a series of messages identified by message ID CPX00I, Comparex lists the keywords in effect for the comparison job at the time of execution. These include any keywords specified explicitly by the user at runtime. For example, we see the following in [Figure 3-2](#):

```
CPX00I - MAXDIFF=5      /* to avoid a runaway compare */
CPX00I - CONTINUE      /* to see final count of differences */
```

The keyword list also includes any installation-specific Comparex defaults that override the normal defaults. These are listed before any runtime keywords that may override them and are set off by a heading bordered by asterisks. For example:

```
CPX00I - *****
CPX00I - ***  I N S T A L L A T I O N  D E F A U L T S  ***
CPX00I - *****
CPX00I -   HALT=COND   /* STOP EXECUTION IF SYNTAX ERRORS FOUND */
CPX00I - *****
CPX00I - ***  E N D  O F  I N S T A L L A T I O N  D E F A U L T S  ***
CPX00I - *****
```

The normal Comparex default value for the HALT keyword as shipped is HALT=NO. But you may customize this default for all reports run at your installation. The difference report lists any customized defaults in effect at the time the job was run — in this example, HALT=COND.

Execution and Output Formatting Parameters

Next, Comparex prints messages showing the execution and output formatting parameters in effect for this job. Execution parameters show the final combination of user-execution settings specified by runtime keywords, custom default settings established for the installation and not overridden by the user, and Comparex defaults not overridden by installation-specific customizations. They also identify the input and output files used in the comparison and whether the files are considered DATA files (fixed format) or TEXT files (free format). Message IDs for this information fall in the range from CPX03I through CPX29I.

Output formatting parameters customize the appearance of the difference report. For example, [Figure 3-2](#) shows the following parameter selections:

```
CPX08I - DECIMAL , EBCDIC , CASE=MIXED , LINE=(32 , HORIZONTAL) , PAGE=58
CPX11I - DASH=C '-' , PLUS=C '+'
```

- DASH=C '-' means that a hyphen will be used to underscore differing bytes.
- PLUS=C '+' means a plus will be used to underscore excess bytes when the SYSUT2 record is longer than the paired SYSUT1 record.
- DECIMAL indicates that the displacement on each line is shown in decimal numbers.

Difference Data

At the top of page 2 of the difference report and on all subsequent pages, Comparex lists the differing records encountered in the compared files and the literals ONE, TWO, or DIFFERENCE in the right-hand column.

- ONE indicates that the line contains data from a SYSUT1 record.
- TWO indicates that the line contains data from a SYSUT2 record.
- DIFFERENCE indicates that a SYSUT2 record has been selected for printing because it differs from corresponding record in SYSUT1.

The amount of detail in the data section of the report depends on the value of the FORMAT keyword. In this example, FORMAT=02 is used, which requests the IBM dump format. In this output format, the SYSUT1 record will display fully, followed by the differing lines of SYSUT2 with the differences underscored.

Summary Statistics

At the end of the report, Comparex prints its "end-of-processing" totals. The format is:

```
SYSUT1(n1)/SYSUT2(n2)SYSUT3(n3),DIFFERENCES(d0,d1,d2)
```

where:

n1 = number of records read from SYSUT1

n2 = number of records read from SYSUT2

n3 = number of records written to SYSUT3

d0 = number of record pairs which differed

d1 = number of records on SYSUT1, not on SYSUT2

d2 = number of records on SYSUT2, not on SYSUT1

For example, [Figure 3-2](#) shows the following summary statistics in message CPX75I:

```
RECORDS PROCESSED: SYSUT1(11)/SYSUT2(12),DIFFERENCES(4,0,1)
EXPLANATION - 4 RECORDS DIFFER THAT SYNCHRONIZED TOGETHER
              0 RECORDS WERE CONSIDERED INSERTED ON SYSUT1
              1 RECORD WAS CONSIDERED INSERTED ON SYSUT2
```

Here *n1*=11, *n2*=12, and because *n2* exceeds *n1* by one, *d2*=1 and SYSUT2 is identified as containing an extra record. If the values of *n1* and *n2* had been reversed, then *d2* would have a value of zero but *d1* would have been set to 1 and SYSUT1 would have been identified as containing an extra record.

The value of *d0* is 4 in this example, because the inserted record on SYSUT2 (after RECORD09) makes all subsequent records unmatched.

What If the Comparison Job Fails?

If the comparison operation runs unsuccessfully, consider the following questions:

- Are your data set names spelled correctly?
- Do the data sets exist?
- Were there any syntax errors (shown in message CPX30A)?

Once you have addressed these questions, rerun the comparison.

DATA Comparison Scenario: KEY Synchronization

Comparex supports three types of synchronization for DATA comparisons:

- KEY synchronization
- SEGMENT synchronization
- Synchronization by physical record number relative to the start of the file

In this next scenario, we will compare the two files you created in [Chapter 1](#) by matching records based on KEY synchronization. Records with the same key value that differ in one or more bytes elsewhere in the record are listed as differing in the difference report.

In addition, for a KEY-synchronized comparison, Comparex detects and writes out-of-sync records to the difference report. Comparex also resynchronizes based on key values after insertions and deletions, so the records following an insertion or deletion do not automatically show as "differences" merely because they are sequenced differently in their files.

Example Job Card Request and Output

The KEY keyword parameter is used to specify to Comparex that records should be matched based on key values before a comparison is made. For the input files you created in [Chapter 2](#), we can use the first eight bytes of each record as a key field. The key values in this field are in character format and ascending sequence. The syntax for the KEY keyword is as follows:

```
KEY=(1,8)
```

where (1) is the starting position of the key field and (8) is the key field length.



NOTE If no key is specified, but the file organization uses a key (such as ISAM or VSAM-KSDS), Comparex will use the primary key specified by the built-in file access method.

Because the previous difference report was in the IBM Dump format, many hexadecimal characters were displayed. In this example, we will restrict the display to alphanumeric characters. To do this, we will specify the following:

```
LINE=80
```

When you created the FILE1 and FILE2 data sets in [Chapter 2](#), was there a sequence number in columns 73 through 80? If so, Comparex probably underscored that number in the difference report shown in [Figure 3-2](#) above. To avoid this in the new difference report, we will specify the following to "mask" out the sequence numbers:

```
MASK=(73,END)
```

We specify specify the MAXDIFF=5 and CONTINUE as before. The resulting job card request syntax is shown in [Figure 3-3](#) below.

Figure 3-3.
KEY Synchronized
DATA Compare
Request

```
//MVS$JOB JOB (accounting,...)
//CPX EXEC PGM=COMPAREX
//SYSPRINT DD SYSOUT=*
//*
//SYSUT1 DD DISP=SHR,DSN=somnode.FILE1
//SYSUT2 DD DISP=SHR,DSN=somnode.FILE2
//SYSIN DD *
MAXDIFF=5 /* to avoid a runaway compare */
```

```

CONTINUE      /* to see final count of differences */
LINE=80       /* Format in alphanumeric only - no HEX characters */
MASK=(73,END) /* Ignore any sequence number in columns 73 - 80 */
KEY=(1,8)     /* KEY will be the field in first 8 characters */
/*

```

The result of the above specifications is to generate a difference report that describes a DATA comparison, pairing records that match on the specified key and isolating inserted and deleted records.

Figure 3-4. The data section of the new difference report should resemble the example below.
KEY Synchronized
DATA Compare
Difference Report

```

C O M P A R E X                                     PAGE 2
SYSUT1=somnode.FILE1,SYSUT2=somnode.FILE2

DSPL |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8

CPX51I - RECORD NUMBER 3 ON FILE SYSUT1
1      RECORD03CCCCCCCCCCCC                       O N E

CPX52I - RECORD NUMBER 3 ON FILE SYSUT2   FIELD=1
1      RECORD03FIRST CHANGE                       T W O
      ----- -DIFFERENCE+

CPX51I - RECORD NUMBER 6 ON FILE SYSUT1
1      RECORD06FFFFFFFFFFFFFF                       O N E

CPX52I - RECORD NUMBER 6 ON FILE SYSUT2   FIELD=1
1      RECORD06SECONDCHANGE                       T W O
      ----- -DIFFERENCE+

CPX36A - KEY OUT OF SPECIFIED SEQUENCE - RECORD 10 ON FILE SYSUT2
1      RECORDXX INSERTION                          T W O

CPX62I - KEY SYNCHRONIZATION MISMATCH - RECORD 10 ON FILE SYSUT2
1      RECORDXX INSERTION                          T W O

CPX62I - KEY SYNCHRONIZATION MISMATCH - RECORD 12 ON FILE SYSUT2
1      RECORD11 INSERTION                          T W O

CPX72I - END OF DATA ON FILE SYSUT2

CPX61I - KEY SYNCHRONIZATION MISMATCH - RECORD 11 ON FILE SYSUT1
1      RECORD26 INSERTION                          O N E

CPX67I - MAXDIFF INVOKED, CONTINUING WITHOUT PRINTING BY REQUEST

CPX71I - END OF DATA ON FILE SYSUT1

CPX74I - BYTES UNDERSCORED(23)

```


CPX75I - RECORDS PROCESSED: SYSUT1(11)/SYSUT2(12),DIFFERENCES(2,1,2)
EXPLANATION - 2 RECORDS DIFFER THAT SYNCHRONIZED TOGETHER
1 RECORD WAS CONSIDERED INSERTED ON SYSUT1
2 RECORDS WERE CONSIDERED INSERTED ON SYSUT2

CPX80I - TIME OF DAY AT END OF JOB: 15:59:59 - CONDITION CODE ON EXIT: 4

Interpreting a DATA Difference Report with KEY, FORMAT, and MASK Keywords

The difference reports in [Figure 3-2](#) and [Figure 3-4](#) compare the same records in the same sequence. The interpretation of these difference reports therefore contains much that is unchanged.

Additionally, however, the difference report generated through KEY synchronization flags out-of-sync conditions. These conditions are reported in the following messages:

CPX36A - The inserted record, RECORDXX, on SYSUT2 is not in the KEY sequence (ascending) specified by default.

CPX62I - The records, RECORDXX and RECORD11, on SYSUT2 are not synchronized to the corresponding records on SYSUT1.

CPX61I - The record, RECORD26, on SYSUT1 is not synchronized with the corresponding record on SYSUT2.

Chapter 4

TEXT Files

TEXT Comparison Summary	27
TEXT Comparison Scenario	27

TEXT Comparison Summary

To Comparex, a TEXT file is any file where no known inter-record relationship exists. TEXT files might have bytes or fields in any fixed relationship within records. Crucially, however, they can contain blanks, may be of variable length, and might be entirely free-format.

TEXT comparison logic compares blocks (or records) of significant data rather than formatted fields. Sequences of blanks of any length are considered equivalent if the significant data on either side of the blank area matches. Excess bytes at the end of a record are not underscored.

TEXT Comparison Scenario

In this scenario we will use TEXT logic to compare the same two files you created in ["Getting Started: Create Two Known Files"](#) on page 18.

Job Card Syntax for a TEXT Compare Request

In the sample job card in [Figure 4-1](#) below, you will recognize the keyword parameters MAXDIFF and CONTINUE from the previous chapter. Two new keyword parameters are used here.

The setting

```
TEXT=JCL
```

tells Comparex to ignore any sequence number and restricts comparison to the first 72 columns of the record — as we might want to do, for example, to compare JCL members.

The PRINT keyword specifies printing a certain number of records from file SYSUT1 as context for each differing record encountered in file SYSUT2. The setting

```
PRINT=MLC
```

specifies a multi-line comparison printout. In such a printout, Comparex prints multiple records before and after the highlighted difference record — a format known as *fade-in, fade-out*. In this example, we accept the default value for the number of MLC lines, which is two.

The resulting job card syntax is shown in [Figure 4-1](#) below.

Figure 4-1. Simple TEXT Compare Request

```

//MVS$JOB JOB ...
//CPX EXEC PGM=COMPAREX
//SYSPRINT DD SYSOUT=*
// PEND
//*
//SYSUT1 DD DISP=SHR,DSN=somnode.FILE1
//SYSUT2 DD DISP=SHIR,DSN=somnode.FILE2
//SYSIN DD *
MAXDIFF=5 /* to avoid a runaway compare */
CONTINUE /* to see final count of differences */
TEXT=JCL /* Compare contents of first 72 bytes only
PRINT=MLC /* Fade-in, fade-out */
/*

```

Example TEXT Difference Report

Figure 4-2. TEXT Compare Difference Report

The difference report generated by the request above should resemble that shown in Figure 4-2 below. The report is interpreted below the figure.

```

RECORD01AAAAAAAAAAAAA                                O N E 1
RECORD02BBBBBBBBBBBBB                                O N E 2

+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7+>+.++++8+++++
D RECORD03CCCCCCCCCCC                                DIF O N E 3      +
-----3-----1-----2-----3-----4-----5-----6-----7-----8-----
I RECORD03FIRST CHANGE                                DIF T W O 3      +
+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7+>+.++++8+++++

RECORD04DDDDDDDDDDD                                O N E 4
RECORD05EEEEEEEEEEEE                                O N E 5

+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7+>+.++++8+++++
D RECORD06FFFFFFFFFFFF                                DIF O N E 6      +
-----3-----1-----2-----3-----4-----5-----6-----7-----8-----
I RECORD06SECONDCHANGE                                DIF T W O 6      +
+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7+>+.++++8+++++

RECORD07GGGGGGGGGGG                                O N E 7
RECORD08HHHHHHHHHHH                                O N E 8
RECORD09IIIIIIIIIII                                O N E 9

+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7+>+.++++8+++++
I RECORDXX INSERTION                                DIF T W O 10     +
+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7+>+.++++8+++++

RECORD10JJJJJJJJJJJ                                O N E 10

+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7+>+.++++8+++++
D RECORD26 INSERTION                                DIF O N E 11     +
-----3-----1-----2-----3-----4-----5-----6-----7-----8-----
I RECORD11 INSERTION                                DIF T W O 12     +
+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7+>+.++++8+++++

CPX71I - END OF TEXT ON FILE SYSUT1
CPX72I - END OF TEXT ON FILE SYSUT2

CPX75I - RECORDS PROCESSED: SYSUT1(11)/SYSUT2(12),DIFFERENCES(3,0,1)
EXPLANATION - 3 RECORDS DIFFER THAT SYNCHRONIZED TOGETHER
0 RECORDS WERE CONSIDERED INSERTED ON SYSUT1
1 RECORD WAS CONSIDERED INSERTED ON SYSUT2

CPX80I - TIME OF DAY AT END OF JOB: 08:52:39 - CONDITION CODE ON EXIT: 4

```

Interpreting a TEXT Difference Report

Records from SYSUT1 that are not matched to any records from file SYSUT2 are identified, on the right-hand side of the report, by **DIF ONE**. Similarly, records from SYSUT2 that are not matched to any record from file SYSUT1 are identified by **DIF TWO**. Furthermore, the logical record number is shown on the far right.

In this scenario, we used the setting PRINT=MLC. If, however, we specified PRINT=FULL, all the records from the file SYSUT1 would have appeared in context.

In this scenario, we chose not to “frame” the differing records. However, had we used the FRAME keyword, the differing records would have been surrounded by the PLUS character and separated by the DASH character.

Chapter 5

Interfaces

This chapter discusses some of the various interfaces to Comparex. Refer to the *Comparex User's Guide* for a complete list of interfaces.

Alternative Comparex Interfaces	31
Generating Delta Decks	31
ISPF Interface	32

Alternative Comparex Interfaces

Comparex interfaces (via CPXIFACE) to many different data collection structures such as CA-Panvalet, CA-Librarian, DL/1, IDMS, , etc. The success of these interfaces depends on the following:

- Your environment
- How CPXIFACE was generated
- How your environment is configured

Additionally, Comparex provides the "Delta Deck Option," enabling you to apply transaction files (commonly called delta decks) against one or more sequential files with the intent of updating them.

Finally, Comparex provides the ISPF interface to help you specify files and keywords.

Generating Delta Decks

This section illustrates how Comparex interfaces to the data structures of ChangeMan ZMF, CA-Panvalet, CA-Librarian, and FUJITSU/FACOM's GEM.

You will recall from [Chapter 1](#) that you can compare one version of a program (at the source level) against another and display the differences. If you were to write those differences to a third file (by specifying COPYDIFF or COPYSAME) and format them appropriately, that file would become a **delta deck**. Comparex has the ability to product delta decks formatted for:

- IEBUPDATE
- MEMBER
- ChangeMan ZMF
- Panvalet

- Librarian (release 3.2 or higher)
- GEM



NOTE Comparex can produce delta decks for ZMF, Panvalet, Librarian, or GEM without those products being installed. However, subsequent application of the delta decks to the original source (the SYSUT1 member) requires the appropriate product.

The following example illustrates how to compare two PDS libraries (PDS1 and PDS2), to create a delta deck (&&DELTA), and to update the first PDS to look like the second PDS. (Refer to the *Comparex User's Guide* for more details on what the delta deck looks like and the functions of CMNDELTA.)

In this example, we use the keyword COPYDIFF to direct Comparex to write the *differing* records from SYSUT2 (as compared to SYSUT1) to SYSUT3. We could have specified the keyword COPYSAME to write the *same* records from SYSUT2 (as compared to SYSUT1) to SYSUT3.

```

//COMPARE EXEC PGM=COMPAREX
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=somnode.PDS1
//SYSUT2 DD DISP=SHR,DSN=somnode.PDS2
//SYSUT3 DD DISP=(,PASS),DSN=&&DELTA,
// UNIT=SYSDA,SPACE=(TRK,(5,5)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6000)
//SYSIN DD *
TEXT=$. /* Wildcard TEXT */
COPYDIFF=(CMN,STAMP=YES) /* ChangeMan ZMF Delta Deck */
/**
//DELTA EXEC PGM=CMNDELTA

```

Figure 5-3. Delta Decks in ChangeMan ZMF Format

ISPF Interface

This interface should simplify the specification of files and keywords for Comparex. It is well suited for the following structures: QSAM, ISAM, VSAM, PDS, Panvalet, or Librarian.

If you are interested in another structure, however, the ISPF interface is probably not your best choice. The reason is because this interface cannot handle the specific allocations of system type libraries required by the more complicated structures.

The ISPF Interface consists of:

- Panel library
- Messages library
- CLIST library (*strictly for installation*)
- Sysgen library
- Load library for modules CPX\$ISPF, CPXPARSE, CMNDELTA, and various utilities
- Table library

Connected to Main Menu

Depending on how the ISPF interface was installed, you might be able to access "CPX@PRIM in 'somnode.COMPAREX.PANELS'" on page 33 directly. More likely you will access the main menu indirectly through a vendor panel similar to the following panel. Serena recommends using a vendor panel to access Comparex.

```

ISRBROBA WSER99.COMPAREX.PANELS(VENDOR) - 08.00      Line 00000000 Col 001 080
Command ==>                                         Scroll ==> PAGE
***** Top of Data *****
)BODY EXPAND(!!)
%!-! Vendor Supplied Software !-!
%Option ==> _ZCMD ! !+
%
% 1 +COMPAREX - Invoke COMPAREX/ISPF
% 2 +Other    - Invoke other vendor software (not supplied)
% X +Exit     - Return to primary option menu
+
+Press%ENTER+to continue; Enter%END+Command to exit.
)PROC
  &ZSEL = TRANS( TRUNC (&ZCMD, '.')
                , '1, 'PANEL(CPX@PRIM) NEWAPPL(CPX) NEWPOOL PASSLIB'
                , '2, '
                , 'X, 'EXIT'
                , '?, '?' )
  &ZTRAIL = .TRAIL
)END
***** Bottom of Data *****

```

Figure 5-4. Vendor Panel

The ISPF interface is *panel driven*. So, from the "VENDOR" panel, you will eventually display the **Comparex Primary Menu** below.

```

CPX@PRIM ---- SERENA (R) COMPAREX (R) / ISPF 8.6 Primary Menu -----
Option ==>                                         Profile: *

0 - SHORT CUT - Single screen for options and dataset names (recommended)
1 - OPTIONS   - Specify compare options for this session
2 - DS NAMES  - Specify dataset names to be compared
3 - SAVE      - Save options profile for future sessions
4 - LOAD      - Select/Delete options profile from prior session(s)
5 - CLEAR     - Clear previously loaded profile
6 - FOREGROUND - Invoke SERENA COMPAREX in the foreground and wait
7 - BACKGROUND - Invoke SERENA COMPAREX as a submitted batch job
8 - BACKGROUND - Similar to above but edit job (and optionally SUBMIT)
9 - PARSE     - Parse Copylib for keyword assistance
X - EXIT      - Exit

Press HELP KEY for tutorial assistance at any point;
Enter END Command to terminate SERENA COMPAREX/ISPF 8.6.0 (2006/105).

SERENA Software                               www.serena.com

```

Figure 5-5. CPX@PRIM in 'somnode.COMPAREX.PANELS'

On panel CPX@PRIM, you can enter an option (*X, or 0 through 9*) or a TSO command like:

```
Option ==> TSO LISTC LEVEL(somnode.COMPAREX)
```

or

```
Option ==> TSO STATUS
```



NOTE You can process TSO commands at any point within the Comparex interface, except for the Tutorial.

DATA Comparison Via Short Cut

For our scenario, we will run a TEXT comparison using the “Short Cut” method. You choose this method of specifying data set names and brief options, by entering **0** at the Option input field on the **Comparex/ISPF Primary Menu**, as follows:

Option ==> 0

Comparex displays the panel CPXSHORT, requesting file names and the necessary free-form keywords.

```

CPXSHORT ----- Dataset Names and Brief Options -----
Command ==>                                     Profile: *

SYSUT1 ISPF FILE:                               SYSUT2 ISPF FILE:
PROJECT ==>                                     PROJECT ==>
LIBRARY ==>                                     LIBRARY ==>
TYPE ==>                                        TYPE ==>
MEMBER ==>                                     MEMBER ==>

OTHER PARTITIONED, SEQUENTIAL, VSAM, OR ISAM DATA SET:
SYSUT1 DSNAME ==> 'USER.SEQ.SYSUT1'
SYSUT2 DSNAME ==> 'USER.SEQ.SYSUT2'

ENTER FREE-FORM KEYWORDS BELOW: (No Syntax Checking Done on the Panel)
====> *<--ASTERISK IN COLUMN 1 DENOTES COMMENT, REMOVE TO USE LINE
====> * TEXT=COBOL,PRINT=MLC,MBRHDR=COND /* COBOL TEXT FADE-IN/OUT
====> * DATA=CSECT,MODE=SYS,MBRHDR=COND /* LOAD MODULES (SYSTEM)
====> KEY=(3,8),FOROUT=(9,GE,C'G2'),MASK=(67,7) /* FLAT DATA

Press Enter to register and stay; enter the END Command to register and exit.

```

Figure 5-6. ISPF Panel CPXSHORT

We will enter the file names in the **OTHER... DATA SET** input area, as follows:

```

OTHER PARTITIONED, SEQUENTIAL, VSAM, OR ISAM DATA SET:
  SYSUT1 DSNAME ==> 'somnode.FILE1'
  SYSUT2 DSNAME ==> 'somnode.FILE2'

```

The listed (commented) free-form keywords under the **ENTER... BELOW** input area are suggestions only. To compare files FILE1 and FILE2, we will apply the keywords used for the scenario in [Chapter 3, "DATA Comparison Scenario: KEY Synchronization" on page 23](#) as follows:

```

ENTER FREE-FORM KEYWORDS BELOW: (No Syntax Checking ...
====> data,key=(1,8) /* Key synchronization
====> field=(1,72),line=80 /* No hex characters
====> maxdiff=5,continue /* Just in case
====>

```

Press **Enter** to register the keywords to ISPF. Then, press **PF3** to return to the main menu, and invoke Comparex as a submitted batch job, as follows:

Option ==> 7

Comparex displays the CPXJOBST panel, enabling you to override the most recent JOB card image.

```

CPXJOBST ----- Job Statement -----
JOB STATEMENT:
==== //COMPARE JOB (X860),'COMPAREX/ISPF', <=== CHANGE ACCORDINGLY
==== //          CLASS=A,_____ <=== CHANGE ACCORDINGLY
==== //          NOTIFY=USER1._____ <=== CHANGE ACCORDINGLY
==== //          MSGCLASS=X_____ <=== CHANGE ACCORDINGLY

Press ENTER to continue;
Enter END Command to terminate JOB submission.

```

Figure 5-7. ISPF Panel CPXJOBST

Make changes to the JOB card (if any) and press **Enter**. A message will overlay the current panel (depending on how the ISPF interface was installed - TSO Submit or Internal Reader), as follows:

```
JOB COMPARE (JOB02598) SUBMITTED
```

This is the response to the TSO Submit command. You will see three asterisks "***" at the bottom of the screen. Pressing **Enter** will return you to the primary menu. The background job you just submitted will run shortly—depending on the machine load.

Index

A

- all-defaults 16
 - about 12
 - mode 10

C

- CA-Librarian 31, 32
- CA-Panvalet 31
- CMNDELTA
 - example 32
- code conventions 7
- Comparex
 - defaults 10
- comparing
 - DATA or TEXT 13
- CONTINUE keyword 27
- COPYDIFF 31
- COPYSAME 31
- CPX@PRIM 33
- CPX00I message 21
- CPX03I to CPX 29I messages 21
- CPX36A message 25
- CPX61I message 25
- CPX62I message 25
- CPXIFACE 16
- CPXJOBST 35
- CPXSHORT 34
- creating an output file 12
- customizing
 - report formats 12

D

- DATA
 - or TEXT comparison 13
- DATA comparison via short cut 34
- DATA files 19
- defaults
 - comparison logic 13
- definitions, environment 13
- delta decks 31
- DIF ONE
 - form of identification 29
- DIF TWO
 - form of identification 29

- DIFFERENCE 21
- difference report 9, 15, 16, 29
- difference reports
 - comparison data 21
 - DATA with defaults 21
 - execution parameters 21
 - keywords 21
 - output formatting parameters 21
 - summary statistics 22
- DL/ 31
- documentation
 - guide to 5
 - online download 6
 - printed 6
 - product CD 6

E

- end-of-processing 22
- environment definitions 13

F

- FACOM 13
- fade-in, fade-out printout 27
- FORMAT 22
- FORMAT keyword 25
- free form keywords 34

G

- GEM 16, 32

H

- how COMPAREX processes information 15

I

- IDMS 31
- IEBUPDATE 32
- interfaces 31
- Introduction 9
- ISAM 32
- ISPF Interface 32

ISPF interface 32

K

KEY 23

KEY keyword 25

KEY synchronization 23

keyword

about 16

keywords

about 12

COPYDIFF 31

COPYSAME 31

difference report listing 21

FORMAT 22

optional 10

L

Librarian 32

libraries

PDS 32

literals

DIFFERENCE 21

ONE 21

TWO 21

M

manuals, information about 5

MASK keyword 25

MAXDIFF keyword 27

MLC (multi-line compare) 27

mode

all-defaults 10

MVS/XA 13

O

ONE 21

Option == => 33

optional keywords 10

OS/390 13

P

panel driven 33

Panels

CPX@PRIM 33

CPXJOBST 35

CPXSHORT 34

Vendor 33

Panvalet 31, 32

PDS 32

PDS libraries 32

PRINT keyword 27

Q

QSAM 32

R

report format

customizing 12

reports

difference 9

difference, TEXT 29

S

same-physical-record-number 23

SEGMENT 23

SYSUT1 22, 29, 32

SYSUT2 29, 32

SYSUT3 32

T

TEXT 34

or DATA comparison 13

TEXT files 27

definition 13

examples 13

TSO Command 33

TWO 21

types of files you can compare

all members 10

CLIST 10

control card images 11

directories 10

documentation 11

executable load modules

CSECT parsing or undefined block to block
10

ISAM 10

job control language (JCL) 10

object code 10

QSAM 10

ranges of members 10

reports 11

selected members 10

sequential 10

source code 10

system intermediate files 10
system transaction files 10
VSAM 10

V

VENDOR 33
VSAM 32

W

what if the comparison job fails?
 considerations 22

Z

z/OS 13

