SERENA®

COMPAREX® 8.7 for z/VSE

Getting Started Guide

Serena Proprietary and Confidential Information

Publication date: April 2011

Table of Contents

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/VSE installations consists of the following manuals:

Manual	Description
Serena® Comparex® for z/VSE Installation Guide	Installation requirements and instructions for installing Comparex under z/VSE.
Serena® Comparex® for z/VSE Getting Started Guide	Quick-start guide to basic Comparex functions.
Serena® Comparex® for z/VSE User's Guide	Comprehensive guide to using Comparex functions in batch and interactive environments under z/VSE.
Serena® Comparex® Quick Reference	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
italics	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.
`monospaced italics`	Indicates names that are placeholders for values you specify; for example, *`filename`*.
`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

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

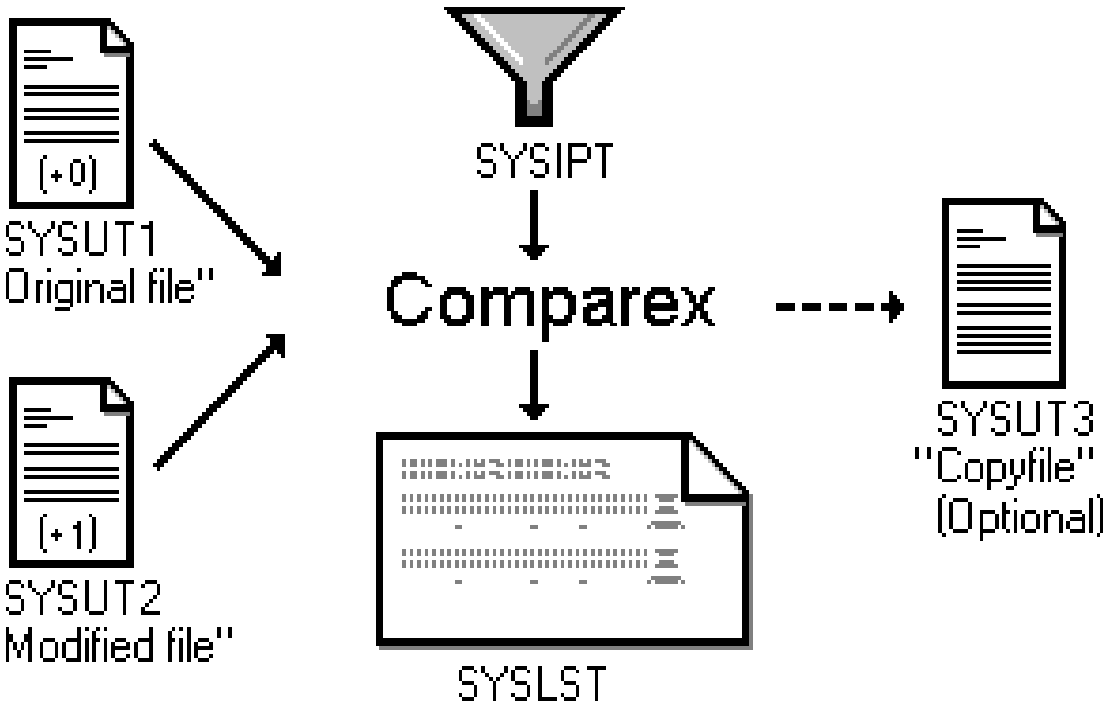
Comparex can compare any two files of almost any structure or organization. Comparex can directly process the following types of files:

- Program
  - Source Code
  - Object Code
  - Executable Load Modules
- Job Control Language (JCL)
- System Master Files
  - Sequential
  - VSAM
- System Intermediate Files
- System Transaction Files
- Directories, Selected Members, Ranges of Members, or All Members
  - CA-Panvalet
  - CA-Librarian
  - z/VSE, VSE/ESA, VSE/SP and DOS/VSE Libraries and Sub-Libraries
  - SLB - Source Statement Library
  - PLB - Procedure Library
  - RLB - Relocatable Library
  - CIL - Core Image Library
  - CA-ROSCOE
  - All types, Fixed, Undefined, Variable, Packed

  - 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
- Members of Core Image Libraries

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

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

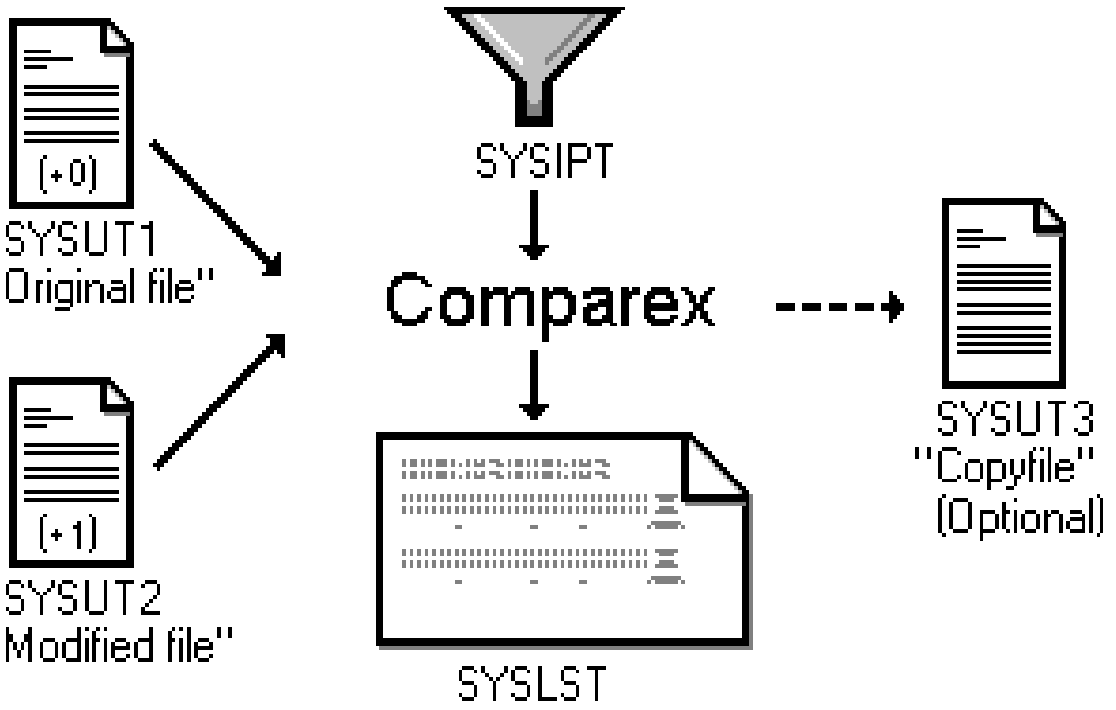
Chapter 2

Getting Started

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* (SYSLST). 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, 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.

SYSIPT As delivered, Comparex operates in an 'all-defaults' mode (see "Running Comparex: All-Defaults or Keyword" on page 17). The most frequently used keywords are already specified. With the optional SYSIPTfile, Comparexenables you to override the default keyword values:

- If SYSIPT is either empty or cannot be opened, Comparex operates with the default values.
- If SYSIPT is not empty, Comparex modifies its default processing with the specified keyword values.

OUTPUT

Comparex generates the following two files:

SYSLST (technically SYS005 but referred to here as SYSLST) 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 COPYSPLIT 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
- Members of Core Image Libraries

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 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/VSE 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.

```
RECORD01AAAAAAAAAAAA
RECORD02BBBBBBBBBBBB
RECORD03CCCCCCCCCCCC
RECORD04DDDDDDDDDDDD
RECORD05EEEEEEEEEEEE
RECORD06FFFFFFFFFFFF
RECORD07GGGGGGGGGGGG
RECORD08HHHHHHHHHHHH
RECORD09IIIIIIIIIIII
RECORD10JJJJJJJJJJJJ
RECORD26 INSERTION
```

```
RECORD01AAAAAAAAAAAA
RECORD02BBBBBBBBBBBB
RECORD03FIRST CHANGE
RECORD04DDDDDDDDDDDD
RECORD05EEEEEEEEEEEE
RECORD06SECONDCHANGE
RECORD07GGGGGGGGGGGG
RECORD08HHHHHHHHHHHH
RECORD09IIIIIIIIIIII
RECORDXX INSERTION
RECORD10JJJJJJJJJJJJ
RECORD11 INSERTION
```

Figure 2-2. Sample Files - FILE1 And FILE2

Note the logical record length *(LRECL)* and block size *(BLKSIZE)* for inclusion into the SYSUT1 and SYSUT2 keywords at execution time.

Chapter 3
DATA File Comparison

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 Chapter 2, "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

```
// JOB    COMPAREX DATA FILES WITH NO KEY
// ASSGN  SYS005,PRINTER
// ASSGN  SYS001,DISK,VOL=WRK001,SHR
// DLBL   SYSUT1,'COMPAREX.FILE1'
// EXTENT SYS001,WRK001
// ASSGN  SYS002,DISK,VOL=WRK002,SHR
// DLBL   SYSUT2,'COMPAREX.FILE2'
// EXTENT SYS002,WRK002
// LIBDEF CL,FROM=SOFTCL
// EXEC   PGM=COMPAREX,SIZE=300K
```

```
  SYSUT1=(DISK,RECFM=FB,LRECL=80,BLKSIZE=400) /* <=== Blksize? */
   DSNUT1=COMPAREX.FILE1
  SYSUT2=(DISK,RECFM=FB,LRECL=80,BLKSIZE=400) /* <=== Blksize? */
   DSNUT2=COMPAREX.FILE2
  MAXDIFF=5    /* to avoid a runaway compare */
  CONTINUE     /* to see final count of differences */
/*
/&
```

Figure 3-2. Simple DATA Difference Report

Submitting the job card above generates output that looks something like Figure 3-2 below:

```
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 - *  KILLRC=YES  /* FORCE RETURN CODE TO BE ZERO          */
CPX00I - ************************************
CPX00I - *** END OF INSTALLATION DEFAULTS ***
CPX00I - ************************************
CPX00I - SYSUT1=(DISK,RECFM=FB,LRECL=80,BLKSIZE=400),DSNUT1=COMPAREX.FILE1
CPX00I - SYSUT2=(DISK,RECFM=FB,LRECL=80,BLKSIZE=400),DSNUT2=COMPAREX.FILE2
CPX00I -  MAXDIFF=5    /* to avoid a runaway compare */
CPX00I -  CONTINUE     /* to see final count of differences */

CPX03I - EXECUTION OF DATANOKY - VALUES EXTRACTED/DEFAULTED:
CPX04I - MAXDIFF=5,CONTINUE,STOPAFT=999999999999

CPX05I - PRINT=(MATCH,MISMATCH),MBRHDR=YES,HALT=COND,KILLRC=NO
CPX06I - WILDCARD=C'.',MODE=APPLICATIONS,(ALL DISPLACEMENTS RELATIVE TO ONE)
CPX08I - DECIMAL,EBCDIC,CASE=MIXED,LINE=(32,HORIZONTAL),PAGE=58
CPX11I - DASH=C'-',PLUS=C'+'
CPX21I - SYSUT1=COMPAREX.FILE1                 DCB=(DISK,RECFM=F,LRECL=80,BLKSIZE=400)
CPX22I - SYSUT2=COMPAREX.FILE2                 DCB=(DISK,RECFM=F,LRECL=80,BLKSIZE=400)

CPX25I - DATA,FORMAT=02
         (FORMAT EXPLANATION: FULL SYSUT1 FOLLOWED BY DIFFERING LINES OF SYSUT2)+
============================================================
C O M P A R E X                                                                  PAGE 2
SYSUT1=COMPAREX.FILE1,SYSUT2=COMPAREX.FILE2

CPX51I - RECORD NUMBER 3 ON FILE SYSUT1

1     D9C5C3D6 D9C4F0F3 C3C3C3C3 C3C3C3C3   C3C3C3C3 40404040 ...     *RECORD03CCCCCCCCCCCC            *  O N E
33    40404040 40404040 40404040 40404040   40404040 40404040 ...     *                                *  O N E
65    40404040 40404040 40404040 40404040                             *                                *  O N E

CPX52I - RECORD NUMBER 3 ON FILE SYSUT2

1     D9C5C3D6 D9C4F0F3 C6C9D9E2 E340C3C8   C1D5C7C5 40404040 ...     *RECORD03FIRST CHANGE            *  T W O
                        -------- ----  --   --------                          ------ -----        -DIFFERENCE+


CPX71I - END OF DATA ON FILE SYSUT1

CPX57I - EXTRA RECORD NUMBER 12 ON FILE SYSUT2
1     D9C5C3D6 D9C4F1F1 40C9D5E2 C5D9E3C9   D6D54040 40404040 ...     *RECORD11 INSERTION              *  T W O
33    40404040 40404040 40404040 40404040   40404040 40404040 ...     *                                *  T W O
65    40404040 40404040 40404040 40404040                             *                                *  T W O

CPX67I - MAXDIFF INVOKED, CONTINUING WITHOUT PRINTING BY REQUEST

CPX72I - END OF DATA ON FILE SYSUT2

CPX74I - BYTES UNDERSCORED(51)
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
CPX80I - TIME OF DAY AT END OF JOB: 15:59:59 - CONDITION CODE ON EXIT: 4
```

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 - *  KILLRC=YES  /* FORCE RETURN CODE TO BE ZERO          */
CPX00I - ***********************************
CPX00I - *** END OF INSTALLATION DEFAULTS ***
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 ombination 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:

```
CPX03I - EXECUTION OF DATANOKY - VALUES EXTRACTED/DEFAULTED:
CPX04I - MAXDIFF=5,CONTINUE,STOPAFT=999999999999
```

- 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

As mentioned in Chapter 1, the three types of synchronization available for DATA comparison logic are:

- KEY
- SEGMENT
- same-physical-record-number (actually, no synchronization)

In this next scenario, we will compare the two files you created in Chapter 1 by matching records based on KEY synchronization. To do this, we will specify the logical key to Comparex as the first eight bytes of each record. (It is in character format and ascending.) The syntax for the KEY keyword is as follows:

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

where (1) is the displacement and (8) is the length.

NOTE If no key is specified, but the file organization uses a key (such as ISAM or VSAM-KSDS), Comparex will use that key.

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 1, was there a sequence number in columns 73 through 80? If so, Comparex probably underscored that number in the difference report. To avoid this, we will specify the following to "mask" out the sequence numbers:

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

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. Through the use of KEYs, this run of Comparex will also detect and write out-of-sync records to the difference report. The output will differ from the previous run because Comparex will now resynchronize after insertions and deletions.

Figure 3-3. KEY DATA Compare Request

```
// JOB    COMPAREX DATA FILES WITH A KEY
// ASSGN  SYS005,PRINTER
// ASSGN  SYS001,DISK,VOL=WRK001,SHR
// DLBL   SYSUT1,'COMPAREX.FILE1'
// EXTENT SYS001,WRK001
// ASSGN  SYS002,DISK,VOL=WRK002,SHR
// DLBL   SYSUT2,'COMPAREX.FILE2'
// EXTENT SYS002,WRK002
// LIBDEF CL,FROM=SOFTCL
// EXEC   PGM=COMPAREX,SIZE=300K
  SYSUT1=(DISK,RECFM=FB,LRECL=80,BLKSIZE=400) /* <=== Blksize? */
   DSNUT1=COMPAREX.FILE1
  SYSUT2=(DISK,RECFM=FB,LRECL=80,BLKSIZE=400) /* <=== Blksize? */
   DSNUT2=COMPAREX.FILE2
  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 */
/*
/&
```

Figure 3-4. KEY DATA Difference Report

Submit the job in Figure 3-3 above. Your output should resemble Figure 3-4 below.

```
SERENA COMPAREX                                                                  PAGE 2
SYSUT1=COMPAREX.FILE1,SYSUT2=COMPAREX.FILE2

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

CPX51I - RECORD NUMBER 3 ON FILE SYSUT1

1     RECORD03CCCCCCCCCCC                                                    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     RECORD06FFFFFFFFFFF                                                    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 Difference Report with KEY DATA

The two difference reports shown in this chapter list the same insertions and deletions. For purposes of presentation, some have been deleted in both Figure 3-2 and Figure 3-4.

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

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 File Comparison

Introduction

To Comparex, TEXT is any file where no known inter-record relationship exists. TEXT files might have bytes or fields in any fixed relationship between records. They can, however, contain blanks and might be entirely free-form.

In this chapter we will use TEXT logic to compare the same two files. You will recognize the keywords MAXDIFF and CONTINUE from the previous chapter. The setting

```
TEXT=JCL
```

lets us ignore the sequence number and use the first 72 columns for the comparison.

The PRINT keyword specifies printing a certain number of records from file SYSUT1 in context with the differing records from file SYSUT2. The setting

```
PRINT=MLC
```

specifies that Comparex will print a few (MLC) records before and after the highlighted differences on the difference report. PRINT=MLC is referred to as *fade-in*, *fade-out*. In this scenario, we have accepted the default value for MLC, 2.

```
// JOB    COMPAREX TEXT FILES
// ASSGN  SYS005,PRINTER
// ASSGN  SYS001,DISK,VOL=WRK001,SHR
// DLBL   SYSUT1,'COMPAREX.FILE1'
// EXTENT SYS001,WRK001
// ASSGN  SYS002,DISK,VOL=WRK002,SHR
// DLBL   SYSUT2,'COMPAREX.FILE2'
// EXTENT SYS002,WRK002
// LIBDEF CL,FROM=SOFTCL
// EXEC   PGM=COMPAREX,SIZE=300K
  SYSUT1=(DISK,RECFM=FB,LRECL=80,BLKSIZE=400) /* <=== Blksize? */
   DSNUT1=COMPAREX.FILE1
  SYSUT2=(DISK,RECFM=FB,LRECL=80,BLKSIZE=400) /* <=== Blksize? */
   DSNUT2=COMPAREX.FILE2
  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 */
/*
/&
```

Figure 4-5. Text File Comparison

Submit the job above. Your output should resemble "Difference Report TEXT=JCL, PRINT=MLC."

```
      RECORD01AAAAAAAAAAAA                                                    O N E 1
      RECORD02BBBBBBBBBBBB                                                    O N E 2

++++++<+++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7+>++.++++8+++++++++++++++
D     RECORD03CCCCCCCCCCCC                                                  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+++++++++++++++

      RECORD04DDDDDDDDDDDD                                                    O N E 4
      RECORD05EEEEEEEEEEE                                                     O N E 5

++++++<+++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7+>++.++++8+++++++++++++++
D     RECORD06FFFFFFFFFFF                                                   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
```

Figure 4-6. Difference Report - TEXT=JCL,PRINT=MLC

Difference Report with TEXT

Because Comparex uses TEXT comparison logic to compare blocks (or records) of significant data rather than fields, Comparex displays an entire record (isolated) without underscoring the differing, or excess, bytes.

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

Introduction

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.

Panvalet and Librarian

Comparex can interface directly to CA-Panvalet or CA-Librarian.

Sample Job for CA-Panvalet

For an example of how to compare two members on the same Panvalet library and generate an audit trail for subsequent browsing or feedback into PAN#1, consider the following:

```
// JOB    COMPAREX PANVALET MEMBERS
// ASSGN  SYS005,PRINTER
// ASSGN  SYS006,DISK,VOL=PAN001,SHR
// DLBL   PANDD1,'somnode.panvalet',,DA
// EXTENT SYS006,PAN001
// ASSGN  SYS003,DISK,VOL=WRK001,SHR
// DLBL   SYSUT3,'TEMP.AUDIT.TRAIL'
// EXTENT SYS003,WRK001,1,0,2000,20
// EXEC   COMPAREX,SIZE=300K
  SYSUT1=(PAN,MEMBER=cobolname1) /* Fill in cobolname1 */
  SYSUT2=(PAN,MEMBER=cobolname2) /* Fill in cobolname2 */
  SYSUT3=(DISK,RECFM=F,BLKSIZE=80),DSNUT3=TEMP.AUDIT.TRAIL
```

```
  TEXT=COBOL,PRINT=MLC,BUFF=500  /* COBOL Text, Fade-in/out */
  COPYDIFF=(PAN,STAMP=YES,TEMP=YES) /* Generate Audit Trail */
/*
/&
*======================> This is the contents of 'COMPAREX.SYSIN.A1'
  SYSUT1=(PAN,MEMBER=cobolname1) /* Fill in cobolname1 */
  SYSUT2=(PAN,MEMBER=cobolname2) /* Fill in cobolname2 */
  SYSUT3='COBNAME1.AUDIT.A1'
  TEXT=COBOL
```

Sample Job for CA-Librarian

For an example of how to compare a member of a Librarian Master against a sequential data set, consider the following:

```
// JOB    COMPAREX LIBRARIAN TO SEQUENTIAL FILE
// ASSGN  SYS005,PRINTER
// ASSGN  SYS006,DISK,VOL=LIB001,SHR
// DLBL   MASTER,'somnode.libarian',,DA
// EXTENT SYS006,LIB001
// ASSGN  SYS002,DISK,VOL=SYSWK2
// DLBL   SYSUT2,'PROGRAM1.ASM'
// EXTENT SYS002,SYSWK2
// EXEC   COMPAREX,SIZE=300K
  SYSUT1=(LIB,MEMBER=PROGRAM1)
  SYSUT2=(DISK,RECFM=FB,LRECL=80,BLKSIZE=800),DSNUT2=PROGRAM1.ASM
     TEXT=BAL,PRINT=MLC
/*
/&
*======================> This is the contents of 'COMPAREX.SYSIN.A1'
  SYSUT1=(LIB,MEMBER=PROGRAM1)
  SYSUT2='PROGRAM1.ASSEMBLE.A1'
     TEXT=BAL,PRINT=MLC
```

NOTE Your Librarian release level must be 3.2 or higher.

z/VSE, VSE/ESA, VSE/SP, and DOS/VSE Libraries

All VSE Libraries, whether in DOS/VSE format (before the VSE/SP 2.1 System Package release) or later formats (VSE/SP, VSE/ESA, or z/VSE) are supported directly.

DOS/VSE Libraries

This is an example of how to compare the directories of an RLB against a CIL.

```
// JOB    COMPAREX DOS/VSE LIBRARY - RELO TO CIL DIRECTORIES
// ASSGN  SYS005,PRINTER
// ASSGN  SYS001,FBA,VOL=VSEPK1,SHR
// DLBL   SYSUT1,'relo.library'
// EXTENT SYS001,VSEPK1
// ASSGN  SYS002,3380,VOL=DOSRES,SHR
// DLBL   SYSUT2,'core.image.library'
// EXTENT SYS002,DOSRES
```

```
// EXEC   PGM=COMPAREX,SIZE=300K
  CPXIFACE=CPXDOSLB        /* Special generation */
*********************************
* Compare the DIRectories only. *
*********************************
  DIR=PDF   /* PDF Format */
  SYSUT1=OTH,DSNUT1=RELO.LIBRARY
  SYSUT2=OTH,DSNUT2=CORE.IMAGE.LIBRARY
/*
/&
```

VSE/SP Libraries - 2.1 Release

Here is an example of how to compare two members of the same type but different sublibraries on different physical libraries.

```
// JOB    COMPAREX VSE/SP LIBRARY MEMBER TO MEMBER
// ASSGN  SYS005,PRINTER
// DLBL   SYSUT1,'private.lib1'
// DLBL   SYSUT2,'private.lib2'
// EXEC   PGM=COMPAREX,SIZE=300K
 TEXT=COBOL,PRINT=MLC   /* COBOL Fade-in/out */
 SYSUT1=(OTH,MEMBER=member1,PARM='ORIGINAL.C')
 DSNUT1=PRIVATE.LIB1
 SYSUT2=(OTH,MEMBER=member2,PARM='SAP40.C')
 DSNUT2=PRIVATE.LIB2
/*
/&
```

DOS 2.1 Libraries — All Members

Here is an example of how to compare the entire contents of sublibrary PROD and type A against sublibrary TEST and type A. We will assume that all members are in Assembler.

```
// JOB    COMPAREX VSE/SP PROD MEMBERS TO TEST MEMBERS (ASM)
// ASSGN  SYS005,PRINTER
// DLBL   SYSUT1,'private.library'
// DLBL   SYSUT2,'private.library'
// EXEC   PGM=COMPAREX,SIZE=300K
  SYSUT1=(OTH,PARM=PROD.A),DSNUT1=PROD.ASM   /* SYSUT1
  SYSUT2=(OTH,PARM=TEST.A),DSNUT1=TEST.ASM   /* SYSUT2
  TEXT=BAL            /* Assume all Assembler text
  PRINT=MLC           /* Fade-in, Fade-=out
  MBRHDR=COND         /* Only see the members that differ
/*
/&
```

Index

K

L

M

O

P

R

S

T

W

Y

Z