

IBM Merge Tool for z/OS and OS/390



User's Guide

Release 1

IBM Merge Tool for z/OS and OS/390



User's Guide

Release 1

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 85.

Third Edition (October 2003)

This edition applies to Version 1 Release 1 Modification Level 1 of IBM Merge Tool for z/OS and OS/390 (program number 5697-H70) and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

This publication is available on the Web at:

<http://www.ibm.com/software/awdtools/sclmsuite/mergetool/>

A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation
H150/090
555 Bailey Avenue
San Jose, CA 95141-1003

or fax your comments from within the U.S., to: 800-426-7773 or, from outside the U.S., to: 408-463-2629.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2003. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this book	v
Who should use this book	v
Releases of Merge Tool supported	v
Required software	v
Prerequisite and related information	v
Summary of Changes - Third Edition (October 2003)	vi

Chapter 1. Product overview	1
Parallel development	1
Parallel development scenarios	1
What does Merge Tool do?	2
Functionality	2
The tools	2
Parallel development solutions with Merge Tool	3
Terminology	4
How Merge Tool works.	5
The merge process	5
The Merge Tool ISPF Workbench	6

Chapter 2. Identify the components to be examined	11
Specify the data set libraries	11
Input libraries	13
Output libraries	14
Output report files	14
Specify the Source Type	15
Specify if member names are the same	17
Building member lists	19
Creating the initial list.	19
Appending to the member list	21
Member lists when the member names are different	23
Primary Commands for the member list.	24
Using Group Actions	25
Include/Exclude actions	26

Chapter 3. Analyze complexity and mark-up changes	29
Merge Tool reports and statistics	29
Generating the reports.	29
Viewing the reports.	30
Report structure	31
Marking up the changes	36
Generating the Work File.	37

Chapter 4. Review changes and address conflicts	39
Displaying the merge components.	39
Interpreting the Work File	42
Line insertion example	42
Line deletion example	43
Line alteration example	45
Conflicting source changes example	46
Editing the Work File or source files	50

Chapter 5. Creating the merged source	51
Generating the Merge File from a Work File	51
Generating the Work and Merge Files	52
Merge outputs	53
Test outputs	53

Chapter 6. The Merge Tool batch utility	55
Merge Tool DD statements	55
Running in batch mode	56
Using batch mode with the default JCL	56
Edit and submit JCL	57
The JCL	59
AZZMRKUP - build a mark-up (Work) file	59
AZZMERG0 - build the final Merge File from the Work File	60
AZZMRGIT - build a mark-up (Work) file and final Merge File	61
The Merge Tool syntax	62

Appendix A. Quick reference for commands	65
Member actions	65
Group actions	66
Primary commands.	66

Appendix B. Sample Merge Tool scenarios	67
Scenario #1 "Basic Parallel Development" The Emergency Production Fix	67
Scenario #2 "Common Parallel Development" Two programmers, one source.	68
Scenario #3 "Complex Parallel Development" Release management in-house	70
Project management actions follow:	70
Programmer actions follow:	71
Scenario #4 "Complex Parallel Development" Vendor customized source	72
Project Management actions follow:	73
Programmer actions follow:	73

Appendix C. The Merge Tool customization process	75
---	-----------

Appendix D. Merge Tool messages	77
--	-----------

Notices	85
--------------------------	-----------

Trademarks.	87
------------------------------	-----------

Bibliography	89
Merge Tool publications	89
Related publications for OS/390	89
Related publications for z/OS	89

Softcopy publications for Merge Tool 89

Index 91

About this book

This book provides guidance and reference information for users of the IBM Merge Tool for z/OS and OS/390. Throughout this book, the term "Merge Tool" refers to the IBM Merge Tool for z/OS and OS/390.

Note: Where the Merge Tool functionality described in this book is new or has changed from that in the previous edition, changes are indicated with revision bars on the left. However, this edition of the User's Guide has been extensively revised to improve the explanations. For this reason, general changes in the documentation have not been marked with revision bars.

Who should use this book

This book is for two kinds of Merge Tool users:

- Application programmers who need to test and debug programs
- Project Managers who need assistance with analysis for determining resources for consolidation efforts.

This book assumes that you are familiar with using ISPF.

To use Merge Tool functions in batch jobs, you must already be familiar with JCL.

Releases of Merge Tool supported

Merge Tool supports Version 1.

Required software

Merge Tool requires the following software:

- OS/390 Version 2 Release 10 or later
- z/OS Version 1 or later

Prerequisite and related information

Merge Tool documentation supports the following tasks:

Planning for, installing, customizing, and maintaining Merge Tool

Refer to the Program Directory (shipped with the product tape) to carry out the SMP/E installation of this product. Then refer to Appendix C, "The Merge Tool customization process," on page 75 for customizing information.

Using Merge Tool

This User's Guide describes the tasks involved in using Merge Tool and gives information about the Merge Tool panels, fields and commands.

For information about related products, see the "Bibliography" on page 89.

Summary of Changes - Third Edition (October 2003)

The following enhancements and changes have been made to Merge Tool for z/OS Version 1 since the last release of the documentation.

- The function of the SS command has been clarified, so that it generates all of the specified statistics reports and no longer generates a Work File.
- Various minor technical corrections.

The User Guide has been completely rewritten, into a task-oriented document. Changes to the technical functionality of Merge Tool are marked in the text by a vertical change bar in the left margin. New, modified and expanded explanations have not been marked.

Chapter 1. Product overview

The IBM Merge Tool for z/OS and OS/390 (Merge Tool) is a source code merging utility that automatically merges up to two versions of source code with the original or base version of source.

Merge Tool is an ISPF application. It uses panels to allow you to select options and to specify parameters, commands and program function (PF) keys to simplify requests for common functions, and full-screen format for information display and editing.

Parallel development

Parallel development is defined as multiple programmers actively working on the same application source concurrently.

Today's applications are complex and project management is challenged to meet objectives and deadlines. Programmers attempting to keep up with the pace often find themselves performing tasks against the same code in parallel with other programmers.

Parallel development scenarios

The types of parallel development are broken into four general types for this manual. They are described below.

Common parallel development

Programmer codes enhancement to the applications main program. While the enhancement is being coded a bug is found and fixed (in the current production source) by the maintenance team.

Basic parallel development

Two programmers are assigned to work on different bugs in the same program. Rather than one programmer waiting until the other programmer has completed their changes, and successfully tested the program, they begin development at the same time.

Release management parallel development

A Company must maintain an existing software release while developing a new release. The maintenance team must continue to support and code against an existing release, while the development team works on a new release of the product.

Vendor application customization

A Company uses a purchased application package (Version 1) from a software vendor. Consultants were hired to make modifications to the software. When the vendor releases a version upgrade (Version 2) the company must assess the resources required for retrofitting the customizations to the upgrade. Once analysis is complete the development team must incorporate their customizations into the upgraded vendor source.

The four scenarios above are very different yet cause many of the same problems.

What does Merge Tool do?

Merge Tool enables both project management and programmers to address the common problems that originate with parallel development, allowing both project management and developers to perform their tasks with the details required and in a timely and accurate manner.

The Merge Tool automatically merges up to two versions of source code with the original or base version of source and produces three major outputs:

1. A marked-up (work) file member is created that shows the results of merging the two versions. It indicates inserted, deleted, altered, and conflicting lines.
2. A merged file member containing all modifications to the source module.
3. Statistics showing the number of modifications, insertions, deletions and conflicts in individual members and summary details showing the number of files processed.

Note: CONFLICTING modifications might require manual intervention.

Functionality

Merge Tool provides the tools necessary to identify, analyze and consolidate independently coded changes that affected the same source.

Merge Tool can:

- Assist project management with analysis, providing the details required for accurately determining what resources are needed for the consolidation effort.
- Assist programmers in identification and owner of changes being merged.
- Assist programmers in isolating CONFLICTING changes and addressing the conflicts.
- Assist programmers in the actual consolidation of several independently coded versions of source.

The tools

Merge Tool provides the tools required by both project management and developers to manage the task of consolidating parallel development. Whether the parallel development effort is of a simple or complex nature, the tasks and tools are the same. What changes is who uses the tools and performs the tasks.

ISPF Workbench

A simple-to-use ISPF front-end allowing for fast access to source members and programmer functions such as editing Work File members. Generated member lists can then be used as the object of the Merge Tool actions and reports in foreground or batch through a single ISPF panel.

Reports

Merge Tool provides reports that assist in the assessment of the effort required for merging the source members. Of particular significance are statistics for conflicting modifications for each member.

The Merge Tool Actions

A variety of actions are available to perform the merge tasks. The user can determine the degree of the merge action based on the complexity of the changes.

- Batch Utility** A batch utility is available to perform most actions that are available in the ISPF Workbench.
- Work File** Output from the initial combination of source. Also known as the marked-up file. The first seven bytes of this file indicate the merge activity that is to take place. This is the sole input to the Merge action.

Parallel development solutions with Merge Tool

Using Merge Tool, the customer can solve many of the parallel development issues.

Common parallel development

Production fix is coded in the same program being upgraded for an enhancement.

Problem

The source for the enhancement does not contain the production fix. Promotion of the enhancement reverses the production fix, causing production failures to return.

Solution

Reconcile the fix into the enhanced version of the source. Prior to promotion the programmer must incorporate the fix into their code using Merge Tool.

Basic parallel development

Two programmers are assigned to work on different bugs in the same program. They begin their assignments concurrently.

Problem

Each version of the fixed program is missing the source for the 'other' fix. Independent promotion of the fixed code causes source regression and production failures to return.

Solution

Programmers must incorporate the independently coded fixes, using Merge Tool. Once incorporated, they must test for both conditions and promote to production.

Release management parallel development

A Company must continue to support and code against an existing release, while the development team works on a new release of the product.

Problem

Because current production source was the base of the new release, eventually the maintenance changes must be incorporated in the next release. Promotion of the new release without incorporating the maintenance performed reverses the production maintenance, causing production failures to return.

Many of the details are unknown. Proper project management requires these details such as number of members, complexity of the changes, and occurrences of conflicting changes when forecasting the resources needed for the reconciliation effort.

Solution

Using Merge Tool, Project Management can assess the development effort required to complete the consolidation of maintenance applied since coding

What does Merge Tool do?

of the new release began. The development team can then reconcile the maintenance into the new version of the source.

Vendor application customization

A Company purchased and customizes a vendor application. The company must assess the resources required for retrofitting the customization to a new release of the application. Once analysis is complete the company must combine their customization with the upgraded vendor source.

Problem: The consultants are no longer on this project. With them went the detailed knowledge of the customization done to this application. The company must determine if there is a conflict between the customization and the changes of the new version and then the customization must be re-coded into the new release of the vendor software.

This scenario is complex. Many of the details are unknown. Accurate project management requires details such as number of members, complexity of the changes, and occurrences of conflicting changes when forecasting the resources needed for the incorporation effort. Additionally, you want to avoid having to entirely re-code the customization.

Solution Using Merge Tool, Project Management can accurately assess the development effort required to address conflicting code changes and complete the retrofitting of the customization with the new release. The development team can then easily reconcile the customization to the new version.

Detailed scenarios and steps are documented in Appendix A.

Terminology

The files of the Merge Tool

Merge Tool uses several files as input and output in the merging process. The files are labeled for ease of identification.

What is a Work File?

The Work File is an intermediate file, also known as the marked-up file. It is the output from the initial combination of source. The merged source is marked-up in the first seven bytes, indicating the final merge activity that is to take place.

The Work File is a PDS and is the sole input to the Merge action.

Merge Tool requires that the Work File be a PDS with an LRECL that is seven bytes larger than the source files used as input to the build Work File process. During foreground processing, if the defined file does not exist, you are prompted with the file allocation panel. In batch execution, you must use the CIGWORK DD statement to identify or create the file.

What is a Base File?

The Base File is the PDS that contains the original version of source

common to both modified versions of source. Both Version1 and Version2 have used the source in this library as the base or starting point for their changes.

What is File1?

File1 is the first of two modified versions of source to be compared and merged with the Base File. This file is required if you wish to create a Work File.

What is File2?

File2 is the second of two modified versions of source to be compared and merged with the Base File.

What is the Merge File?

The Merge File is the output from the Merge step. The Merge step reads the Work File and, based on the mark up indicators, builds a combined source. This source can be reviewed and then added to the site Source Management Tool, such as SCLM, creating output that can be tested for the functions of both File1 and File2 modifications.

What is the Log File?

Merge Tool uses the Log File to reflect any parameters and statements in effect during any of the merge processes. If there is an error in the processing, the reason is documented in this file.

This file can be a sequential data set, SYSOUT, or a partitioned data set.

What are the Statistics Files?

Merge Tool writes statistical data to two files.

Member statistics are written to the CIGSTATS DD statement. These statistics include number of lines inserted, deleted and any conflicting changes.

Summary statistics are written to the CIGSUMM DD statement. These statistics include totals related to number of members processed.

Other terms

Conflict

The result of two programmers changing the same line or inserting different lines in exactly the same place in their particular versions of a program. Conflicting source is identified in the Work File and is reported on in the Member Statistics.

Overlap Area

The location in the Work File source where conflicting changes occurred.

How Merge Tool works

The merge process

The basic steps for resolving parallel development issues are:

1. Identify the components to be examined.
2. Analyze the complexity of the project and mark up the changes.
3. Review the marked-up changes and address conflicts.
4. Create the merged source.
5. Test the outputs from the merged source.

How Merge Tool works

All but the last (testing) step can be performed directly from within Merge Tool. The steps are described below.

Identify the components to be examined

Working with Merge Tool, you can create a list of data set members in which the File1 members are matched up with the Base File and File2 members.

Analyze the complexity of the project and mark up the changes

You can get Merge Tool to generate statistics on individual member or summary statistics, showing information such as the number of inserts, deletes and conflicting changes that occurred in the members. Based on the statistics, the user can get a better understanding of the resources required for the merge process.

You mark up the changes that have occurred in your members, by generating a Work File that compares the base program and one or two files that originated from the base file. In the Work File, Merge Tool places *merge directives* in the first seven bytes of each line to document the insertions, deletions, and overlap areas. Merge directives are used to determine what actions the merge step takes for that line of source.

Review changes and address conflicts

Via Merge Tool, you can view and edit the source or Work Files to resolve conflicting change issues. You can try different scenarios by changing the merge directives.

Create the merged source

In this step, you get Merge Tool to generate a Merge File from the mark-up directives in the Work File. The Merge File is source that can be stored in the site Source Management Tool, such as SCLM, or used as input to a compiler. The Work File is the sole input to the merge step. Merge Tool reads the input and performs the insertions and deletions indicated in the first seven bytes of each line in the Work File.

Test Outputs from merged source

The final step is to add the merged source into the site source management repository, such as SCLM, creating outputs that can be tested for all functionality. If the testing is unsuccessful, you can perform steps 3 through to 5 repeatedly until successful testing is achieved. At this time, Merge Tool does not interface with or automatically add the merged source into any source management tool.

The Merge Tool ISPF Workbench

The Merge Tool Workbench is a set of ISPF panels that enable you to work with Merge Tool interactively. Access to the Workbench is usually provided through the ISPF Primary Options Menu panel, although your systems programmer might have provided an alternative pathway. See Appendix C, "The Merge Tool customization process," on page 75 for details.

Figure 1 on page 7 shows Merge Tool's main panel, as it appears when first accessed.

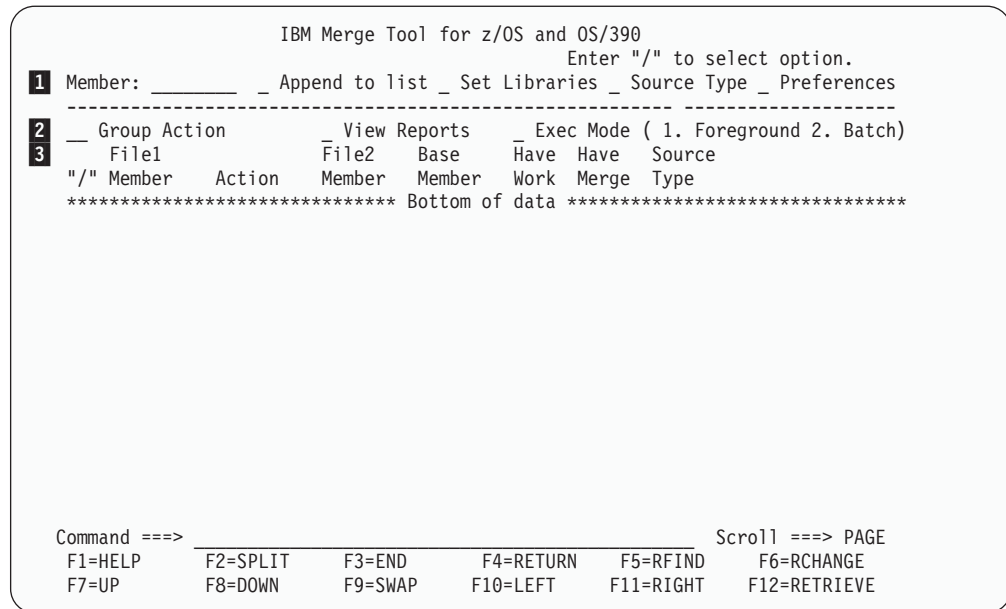


Figure 1. Merge Tool Main Panel

The main panel is broken into three sections:

1 Session information

The Set Libraries, Source Type and Preferences fields in this section are used to determine the session settings. Once set, these settings are retained between ISPF login sessions or until altered by the user. The Member and Append to list fields are used to build a list of members to be acted upon within the current session. Values in these fields are not retained between sessions.

2 Action information

The fields in this section are used to specify the actions to be taken on members in the member list section.

2 Member list section

This section contains the column headings for the member information. Once a member list has been built, the data is displayed in the area below the headings.

Panel fields

Table 1. Description of Fields on Main Panel

Field	Description
Member	Specify a member name or a member pattern to work with specific members or '*' for all members. You can use a "/" in the member field to access the Specify Members panel.
Append to List	If selected, indicates that you want to add members to your current member list. Leave blank, if you want to create a new member list when specifying a member name or pattern. Enter a "/" to append multiple query requests to the displayed list of matching members.
Set Libraries	If selected, displays a panel in which you can specify libraries or change your current libraries settings.

How Merge Tool works

Table 1. Description of Fields on Main Panel (continued)

Field	Description
Source Type	If selected, displays a panel in which you can specify the Source Type for the members being added to the current member list. Source Type is used by Merge Tool to determine where line numbers are, so that they can be ignored. It is also needed in order for Merge Tool to be able to ignore COBOL, ASM, or JCL comments.
Preferences	If selected, displays a panel in which you can set session preferences, such as using View or Browse mode when editing.
Group Action	Enter "/" to see group action choices. A group action is applied to all non-excluded members in the current member list.
View Reports	View current report files.
Exec Mode	Enter to "1" to execute in foreground; enter "2" to execute in batch mode.

Member List section

Once a list of members has been built, the data is displayed beneath the column headings. For descriptions of each column heading, see "Interpreting the member list" on page 20.

```

                                IBM Merge Tool for z/OS and OS/390      Row 1 to 7 of 7
                                Enter "/" to select option.
Member: _____ / Append to list _ Set Libraries _ Source Type _ Preferences
-----
_ Group Action      _ View Reports      1 Exec Mode ( 1. Foreground 2. Batch)
_ File1            File2   Base       Have Have   Source
"/" Member   Action   Member   Member   Work Merge  Type
_ AZZASM01      YES    YES        YES        YES        ASM___
_ AZZCBLA1      YES    YES        YES        YES        COBOL__
_ AZZCBLA2      YES    YES        YES        YES        COBOL__
_ AZZCBL01      YES    YES                YES        COBOL__
_ AZZCBL02      YES    -----                YES        COBOL__
_ AZZCBL11      AZZCBL03 YES                YES        COBOL__
_ AZZCBL01      ----- YES                YES        COBOL__
***** Bottom of data *****

Command ==> _____ Scroll ==> PAGE
F1=HELP      F2=SPLIT      F3=END        F4=RETURN     F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP       F10=LEFT      F11=RIGHT     F12=RETRIEVE

```

Figure 2. Main Panel showing data in the Member List section

Merge Tool usage notes

The member list is compiled from the File1 data set members. Only members in File1 that match the pattern supplied in the Member field are displayed in the list.

When the member being merged is not found in the base file, all the lines of File1 and File2 are merged. All lines are considered inserts.

Error messages

Merge Tool supplies a short message in the upper right hand corner of the panel if an error is detected. The user can request a more detailed message by pressing PF1. A long message is then displayed in a pop-up window.

Critical information

You are strongly advised to turn off the Function Key display off, when using Merge Tool. Some error messages cannot be seen when the Function Keys are displayed. To turn the display off, enter PFSHOW on the Command Line. (PFSHOW is a toggle switch).

```

IBM Merge Tool for z/OS and OS      No member(s) found
                                   Enter "/" to select option.
Member: ZZZ_____ / Append to list _ Set Libraries _ Source Type _ Preferences
-----
  Group Action      _ View Reports      1 Exec Mode ( 1. Foreground 2. Batch)
  File1            File2 Base      Have Have Source
"/" Member      Action      Member Member Work Merge Type
-----
  AZZASM01        YES      YES      YES      ASM_____
  AZZCBLA1        YES      YES      YES      COBOL____
  AZZCBLA2        YES      YES      YES      YES      COBOL____
  AZZCBL01        YES      YES      COBOL____
  AZZCBL02        YES      ----- COBOL____
  AZZCBL11        YES      YES      COBOL____
  AZZCBY01        ----- YES      COBOL____
***** Bottom of data *****



No members found for the member name you entered, check libraries



Command ==> _____ Scroll ==> PAGE
    
```

Figure 3. Error Messages

Merge Tool HELP facility

The Merge Tool ISPF Workbench has a full suite of Tutorial Panels to assist the user. The user can press PF1 or enter "HELP" at the command line on any panel or pop-up to get information about how to use that particular panel.

Chapter 2. Identify the components to be examined

The first step in the process of resolving parallel development issues is to identify the components that need to be examined. Using Merge Tool, you can build up a list of members for which the development paths have diverged and, from that, determine a list of merged source members that need to be created.

For each member, you must identify the merge components and specify the source type and other factors. The merge components are the data set libraries that hold the Base, File1 and File2 members that might have been modified and the data set libraries in which you store the generated Work File, Merge File and various report and statistics files.

Once your list of members has been generated, you can perform analysis or merge actions on those members individually or as a group.

The overall process needed to build up your list of members is to:

1. Specify the data set libraries in which the input merge components are found and output merge components are stored.
These libraries can be changed for each member or group of members to be added to the list.
2. Specify the Source Type of the input merge components.
This Type can be changed for each member or group of members to be added to the list, but must be the same for File1, File2 and Base.
3. Specify whether or not the member names for File1, File2 and Base are the same.
4. Specify the full name or name pattern of the members to be added to the list.
5. Specify whether you are starting a new list or appending the names to the current member list.
6. Generate the initial list, then append any members with differing specifications.

The following sections describe each task in detail.

Specify the data set libraries

The Set Libraries pop-up panel is used to specify the partitioned data sets that Merge Tool uses as components in the merge process.

To specify your libraries:

1. On the main panel, type a "/" into the Set Libraries field.
2. Enter a 1 (Foreground) into the Exec Mode field.

Note: Exec Mode only affects actions performed against members in the member list area and has no effect upon the session settings. However, because it is a compulsory field, it must be completed so that the Merge Tool can process the current panel and display the selected session setting panel.

3. Press Enter. The Merge Tool Set Libraries panel (Figure 4 on page 12) is displayed.

Specify the data set libraries

This panel shows the current session settings for these libraries. If this is the first time that Merge Tool has been used, all fields are empty.

```

M                                     Merge Tool Set Libraries
-                                     -----
-                                     Input Libraries (File1, File2 and Base1 Required)
"                                     -----
*                                     Batch)
                                     -----
                                     *****

                                     Output Libraries (Required)
Work..... 'USERID.SAZZSAMW'
Merge..... 'USERID.SAZZSAMM'

                                     Output Report Files (Optional)
Member Stats.. 'USERID.SAZZSAML'
                                     Disp: _mod
Summary Stats. 'USERID.SAZZSAMS'
                                     Disp: _mod
Log..... 'USERID.SAZZSAMZ'
                                     Disp: _mod

C Command ==>                                     PAGE
```

Figure 4. Set Libraries

4. For each report that you want to receive, type the data set name into the appropriate field.
5. For each report specified in the previous step, select (/) or clear the Disp: _mod field as required. This field determines whether the generated reports replace any existing data in the data set (cleared) or are appended to the existing data (selected).
6. Press Enter.

If all of the specified data set names exist, the main panel is displayed with a confirmation message in a pop-up window.

```

IBM Merge Tool for z/OS and OS/390
Update Completed
M
- SET_LIBRARIES      : UPDATE COMPLETED!
-
- TO APPLY THESE CHANGES YOU MUST REBUILD THE LIST
"
* BE SURE TO UPDATE THE APPROPRIATE FIELDS:
  (APPEND TO LIST, MEMBER ETC.)
      ption.
      erences
      -----
      . Batch)
      *****

Command ==> _____ Scroll ==> PAGE

```

Figure 5. Set Libraries confirmation message

If one or more of the data set names specified for the Input Libraries does not exist, an error message is displayed. You must change the name to that of an existing data set and press Enter again.

If one or more of the data set names specified for the Output Libraries or Output Report Files does not exist, you are prompted to allocate the data set. (See Figure 6 on page 14 and Figure 7 on page 15 for examples of the allocation panels. Once all allocations have been specified, the main panel is displayed with a confirmation message in a pop-up window.

Note: These settings only come into effect the next time that the list is built or altered. They are not applied to existing list members.

Input libraries

The input libraries specify the partitioned data sets that contain the modules that you want to merge.

Table 2. Input Libraries

Library	Description
File1	Specify the name of a library that contains modified source modules.
File2	Specify the name of another library that contains modified source modules. Note: If this field is set to the same library as File1, the File1 and File2 members must have different names. You need to set up your preferences to reflect this (see "Specify if member names are the same" on page 17).
Base	Specify the name of the library that contains the original source modules used to create the "File1" and "File2" members. Up to four libraries can be specified. The libraries are searched in the order specified on the panel.

Specify the data set libraries

Output libraries

The output libraries are needed for the marked-up (Work) file and the merged source modules.

Table 3. Output Libraries

Library	Description
Work	Specify the name of a partitioned data set where the intermediate merged file member is to reside. The data set must have an LRECL (Logical Record Length) that is at least 7 bytes longer than the File1 LRECL. If the data set does not exist, Merge Tool prompts you to allocate it. You can let Merge Tool automatically allocate the space or you can specify the allocation spacing yourself.
Merge	Specify the name of a partitioned data set where the final merge output is to reside. If the data set does not exist, Merge Tool prompts you to allocate it. You can let Merge Tool automatically allocate the space or you can specify the allocation spacing yourself.

Figure 6 is an example of the Allocate panel that Merge Tool displays when an Output Library data set does not exist.

```
IBM Merge Tool for z/OS and OS/390
Allocate Library
File for MERGE library does not exist:
'USERID.TESTMRG'
To allocate it with same characteristics of
'USERID.SAZZSAM1'
Optional, specify space:
Tracks _ or Cylinders _
Primary Space ____ Secondary Space ____
Directory Blocks ____
Optional, select another volume:
Volume ____
Press Enter to Allocate or PF3 to cancel.
Command ==> _____
Command ==> _____ Scroll ==> PAGE
```

Figure 6. Allocate Library

Merge Tool offers to allocate the new data set with the same spacing characteristics as used in File1. Alternatively, you can specify space by selecting either tracks or cylinders, primary space, secondary space and directory blocks. You can select the volume, by entering a desired vol-ser. The data set is allocated with the appropriate DCB attributes (LRECL, RECFM and BLKSIZE) for a Work or Merge File.

Output report files

The output report files are optional. You specify flat files or partitioned data sets if you want to route the reports produced by Merge Tool to files.

When you execute the Merge Tool Member Actions in Foreground mode, you need to specify files in order to view the member statistics and the summary statistics.

When you are running the batch utility, the reports are routed to SYSOUT if you do not specify a file.

Table 4. Output Report Files

Report File	Description
Member Statistics	Number of modifications for each source module processed. Statistics showing conflicting modifications are of particular significance. If there are conflicting changes, you probably need to manually intervene, making sure the merged source module functions as intended.
Summary Statistics	Totals related to number of members processed.
Log	Parameters and statements in effect during the merge process. If processing does not complete successfully, the reason is reported here. If this is left blank and processing is performed in foreground, then the log file is not produced.

Figure 7 is an example of the Allocate panel that Merge Tool displays when an Report data set does not exist.

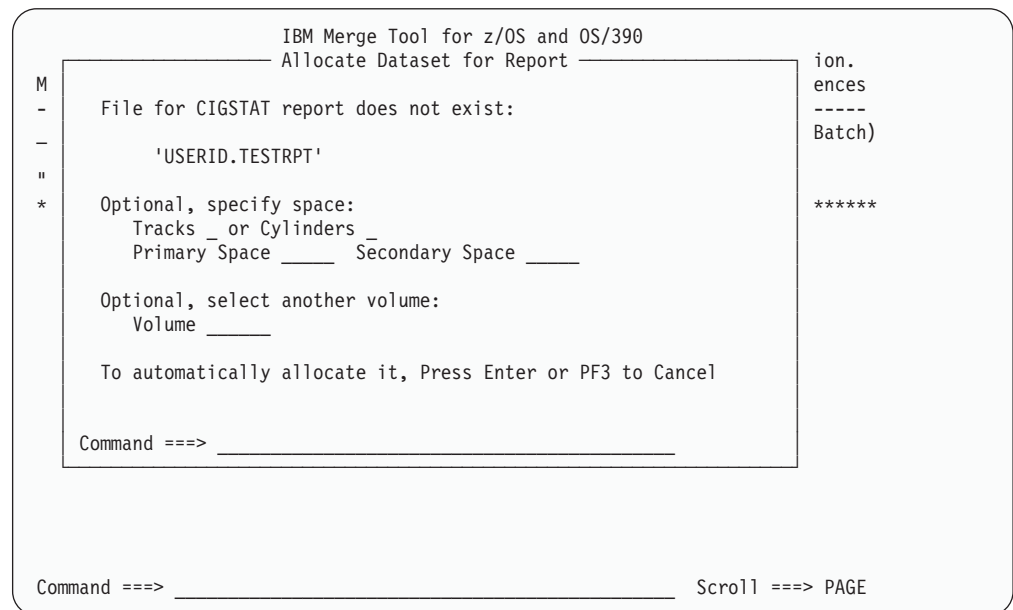


Figure 7. Allocate data set for report

Merge Tool offers to allocate the new data set with its default spacing values or you can specify space by selecting either tracks or cylinders, primary space, secondary space and directory blocks. You can select the volume, by entering a desired vol-ser. The data set is allocated with DCB attributes (LRECL, RECFM and BLKSIZE) appropriate for a Report file.

Specify the Source Type

The Source Type is used by Merge Tool to determine where within the record length the actual code resides, so that line numbers can be ignored in a comparison. Merge Tool can be used to compare any text source data, including C++ and Java code. You can choose a built-in source type for COBOL, ASM, JCL, DATA and PL1. For other source types, you must specify the column range to be used in the comparison.

Specify the Source Type

To specify your Source Type:

1. On the main panel, type a "/" into the Source Type field.
2. Enter a 1 (Foreground) into the Exec Mode field.

Note: Exec Mode only affects actions performed against members in the member list area and has no effect upon the session settings. However, because it is a compulsory field, it must be completed so that the Merge Tool can process the current panel and display the selected session setting panel.

3. Press Enter. The Specify Source Type panel (Figure 8) is displayed.

This panel shows the current session settings for the Source Type. If this is the first time that Merge Tool has been used, all fields are empty.

The screenshot shows a terminal window titled "IBM Merge Tool for z/OS and OS/390" with a sub-panel titled "Specify Source Type". The sub-panel contains the following text: "Specify Type: COBOL__ (COBOL, ASM, JCL, DATA, PL1)", "COLUMNS FOR COBOL ARE 7:72", "or", "Specify compare column start ____ and end ____", and "Command ==> _____". To the right of the sub-panel, there is a menu with options: "ect option.", "Preferences", "-----", "und 2. Batch", and "*****". At the bottom of the terminal window, there is a "Command ==> _____" prompt and a "Scroll ==> PAGE" indicator.

Figure 8. Specify Source Type

4. To use the default column start and end values for one of the built-in Source Types, enter the type's initial or full name in the Specify Type field.

C - COBOL

The code start and end columns are set to 7:72. Line numbering is assumed to be in the COBOL sequence number area (columns 1 - 6).

A - ASM

The code start and end columns are set to 1:72. Line numbering is assumed to be in the ISPF Edit line number area (columns 73 - 80).

J - JCL

The code start and end columns are set to 1:72. Line numbering is assumed to be in the ISPF Edit line number area (columns 73 - 80).

D - Data

The code start and end columns are set to 1:133. It is assumed that there is no line numbering.

P - PL1

The code start and end columns are set to 1:72. Line numbering is assumed to be in the ISPF Edit line number area (columns 73 - 80).

Specify the Source Type

Alternatively, you can specify the location of the code to be compared, by entering values in the Specify compare column start and end fields.

Note: You cannot enter values in both the Specify Type and the Specify compare column fields. These fields are mutually exclusive.

5. Press Enter.

If the values entered are valid, the main panel is displayed with a confirmation message in a pop-up window.

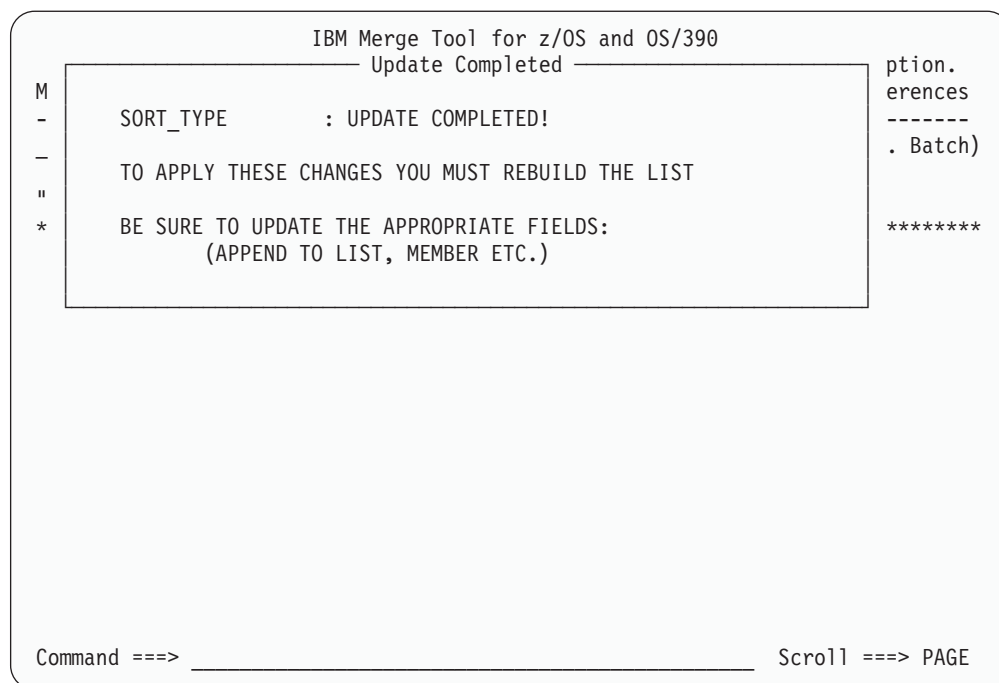


Figure 9. Set Source Type confirmation message

If the values entered are not valid, an error message is displayed in the upper right corner of the screen. Correct the problem and press Enter again.

Specify if member names are the same

You must determine if the PDS member names are the same among the three files, Base, File1 and File2. If these members are in the same data set, the names must be different. For example, you might have used a member name mask, such as programmer initials, that prefixes the actual member name. If the corresponding members are in different data sets, the names for each component can be identical or different, depending upon your site's conventions.

Before adding members to the member list, you must tell Merge Tool whether or not your matching Base, File1 and File2 members have the same name. To do this:

1. On the main panel, type a "/" into the Preferences field.
2. Enter a 1 (Foreground) into the Exec Mode field.

Note: Exec Mode only affects actions performed against members in the member list area and has no effect upon the session settings. However, because it is a compulsory field, it must be completed so that the Merge Tool can process the current panel and display the selected session setting panel.

Specify if member names are the same

3. Press Enter. The Merge Tool Option Default Settings panel (Figure 10) is displayed.

This panel shows the current session settings. If this is the first time that Merge Tool has been used, all fields are empty.

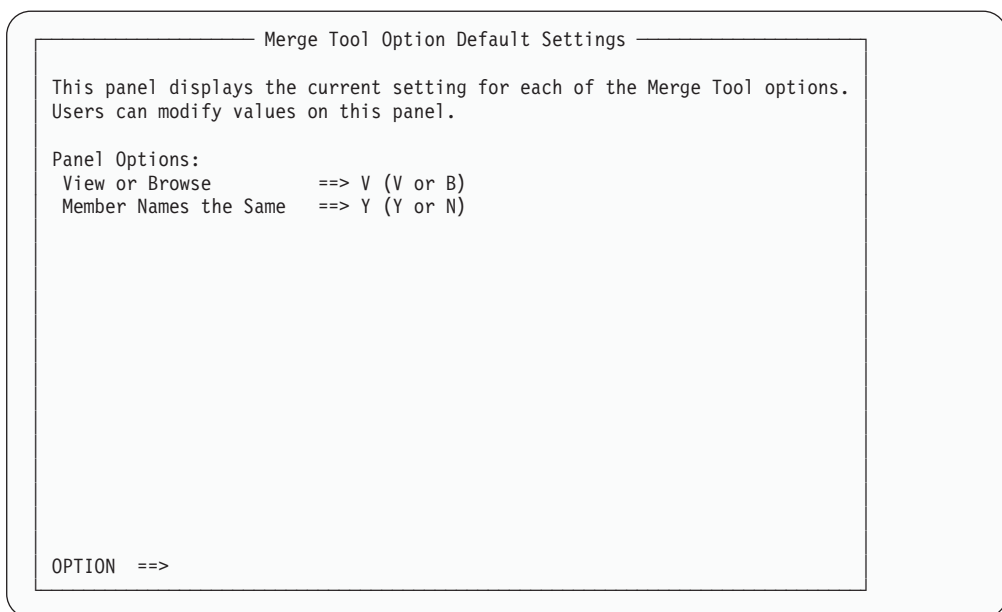


Figure 10. Option Default Settings

4. Type Y or N into the Member Names the Same field, as appropriate.

Y - Yes

The member names are the same in each data set. Selecting this option implies that you have used different data set libraries to store your merge components. When adding members to your member list (see “Building member lists” on page 19), Merge Tool matches the members in each of the specified libraries, on the basis of identical member names.

Note: If you have specified the same data set for each merge component library *and* set the member names to be the same, Merge Tool matches each file with itself and any subsequent comparisons report no changes in the member.

N - No

The member names are different in each data set. Selecting this option implies that you have used the same data set library to store your merge components, however, this is not necessarily true. When adding members to your member list (see “Member lists when the member names are different” on page 23), Merge Tool prompts you for the names of the members to be used as the Base, File1 and File2 components. It uses the file (or files) specified in Set Libraries as a default location for these members but gives you an opportunity to specify alternative data sets in which each member can be found.

5. Press Enter.

The main panel is displayed with a confirmation message in a pop-up window.

```

IBM Merge Tool for z/OS and OS/390
Update Completed
M
-  PREFERENCES      : UPDATE COMPLETED!
-
-  YOUR PREFERENCE FIELDS HAVE BEEN UPDATED
"
*
      ption.
      erences
      -----
      . Batch)
      *****

Command ==> _____ Scroll ==> PAGE

```

Figure 11. Update Preferences confirmation message

Note: Figure 10 on page 18 also shows the View or Browse option. This option controls the way in which members and merge components in a list are opened for examination. See “Displaying the merge components” on page 39 for details.

Building member lists

Once the preferences and libraries have been set for the query, you are ready to build a list of members for the merge process. You build your list by creating an initial list of members and then appending any additional members. Each time that you add new members to the list, you can change the merge component libraries, the source type and the member name preference. If you have specified that the members do not have the same name, you also need to match the member components manually.

Creating the initial list

To build an initial list of members:

1. Ensure that your merge component libraries, source type and member name preferences have been customized to suit the first member or group of members to be added to the list.

Note: This section describes the process used when your member name preference has been set to Yes. For details about adding members where the names are not the same, see “Member lists when the member names are different” on page 23.

2. Ensure that the Append to list field is cleared. The list that you generate replaces any existing list members.
3. Enter a 1 (Foreground) into the Exec Mode field.

Note: Exec Mode only affects actions performed against members in the member list area and has no effect upon creating a member list.

Building member lists

However, because it is a compulsory field, it must be completed so that the Merge Tool can process the current panel and display the specified members.

- In the Member field, type either the full member name or a name mask pattern, such as AZZCBL*. To include all members from the File1 data set, type a * in the field.

The member list is compiled from the File1 data set members. Only members in File1 that match the pattern supplied in the Member field are displayed in the list.

- Press Enter. A list of members is displayed in the main panel.

```

                                IBM Merge Tool for z/OS and OS/390          Row 1 to 7 of 7
                                Enter "/" to select option.
Member: _____ _ Append to list _ Set Libraries _ Source Type _ Preferences
-----
_ Group Action          _ View Reports      1 Exec Mode ( 1. Foreground 2. Batch)
_ File1                File2 Base         Have Have Source
"/" Member Action      Member Member      Work Merge Type
_ AZZCBLA1             YES YES            YES YES COBOL_
_ AZZCBLA2             YES YES            YES YES COBOL_
_ AZZCBL01             YES YES            YES YES COBOL_
_ AZZCBL02             YES -----        YES YES COBOL_
_ AZZCBL11             YES YES            YES YES COBOL_
_ AZZCBY01             ----- YES         YES YES COBOL_
***** Bottom of data *****

Command ==> _____ Scroll ==> PAGE

```

Figure 12. Member List

Interpreting the member list

The following table describes the fields seen in the member list area.

Table 5. Member List Section

Heading	Description
"/"	The label for the Line Command fields (___). You can enter a "/" character into any of the fields in the column to view a list of possible Line Commands, or you can enter the command directly in the field. See "Generating the Work File" on page 37 and Appendix A, "Quick reference for commands," on page 65 for details about using this field.
File1 Member	The name of a member in the library specified for File1.
File2 Member	An entry indicates if there is a corresponding member in File2. If members are named the same and a corresponding member exists, it shows "YES". If a corresponding member does not exist, it shows "-----". If members are not named the same and a corresponding member exists, it shows the corresponding member name. If a corresponding member does not exist, it shows "-----".

Table 5. Member List Section (continued)

Heading	Description
Base Member	An entry indicates if there is a corresponding member in the Base file. If members are named the same and a corresponding member exists, it shows "YES". If a corresponding member does not exist, it shows "-----". If members are not named the same and a corresponding member exists, it shows the corresponding member name. If a corresponding member does not exist, it shows "-----".
Have Work	An entry indicates whether or not a marked-up file exists in the library specified as the Work File PDS for that member. The Work File member name would be the same as the member name of File1.
Have Merge	An entry indicates whether or not a merge file exists in the library specified as the Merge File PDS for that member. The Merge File member name would be the same as the member name of File1.
Source Type	Indicates the source type of the file. If you have specified a source type on the "Source Type" pop-up panel, it is carried to this column. You can over-type the source type for an individual member. You can also specify a start and end column to be used during the merge process instead of specifying a source type.

When attempting to merge entire libraries, Merge Tool assists you in analyzing the project complexity by identifying, at a glance, which File1 members have matched members in the Base File and File2 libraries. In Figure 12 on page 20 there are four members that have a File1, File2 and Base member match.

Based on the type of research you are performing, the missing members might indicate that out of all the members in File1, only four need to be merged with File2 and the Base File member.

Appending to the member list

When you change the content of the Member field and press Enter, Merge Tool assumes that a new list is being created. Merge Tool replaces the contents of the current member list with the results of the new query.

When you want to add new members but wish to keep the current list, you must request that Merge Tool *append* the new results to the current list. This is done by selecting the "Append to list" field. You can also specify a different set of libraries and source types for the appended members. For example, you might have stored all of your COBOL source members in one group of data set libraries and now want to add an ASM (Assembler) source member which has been stored in a different location.

The following example shows the main panel settings needed to append a member from a different library, with a different source type. Note that the Append to list, Set Libraries and Source Type options have all been selected.

Building member lists

```
IBM Merge Tool for z/OS and OS/390      Row 1 to 6 of 6
Enter "/" to select option.
Member: AZZA*__ / Append to list / Set Libraries / Source Type _ Preferences
-----
  Group Action      _ View Reports   1 Exec Mode ( 1. Foreground 2. Batch)
  File1           File2   Base   Have   Have   Source
"/" Member      Action  Member Member Work Merge Type
-- AZZCBLA1      YES    YES    YES    YES    COBOL__
-- AZZCBLA2      YES    YES    YES    YES    COBOL__
-- AZZCBL01      YES    YES                    COBOL__
-- AZZCBL02      YES    ----- COBOL__
-- AZZCBL11      YES    YES                    COBOL__
-- AZZCBL01      ----- YES                    COBOL__
***** Bottom of data *****
Command ==> _____ Scroll ==> PAGE
```

Figure 13. Main Panel before append

When Enter is pressed, Merge Tool displays the Set Libraries panel first. In this panel, you would enter the data set names of the libraries in which your ASM member components can be found (and where you would like the output stored, if also different) and then press Enter again.

Merge Tool displays the Source Type panel next, in which you can change the source type to ASM and then press Enter again.

Merge Tool now processes the pattern in the Member field and appends the member to the list. In doing this, Merge Tool retains the links between each member and its component libraries and source type. In other words, it still "remembers" the libraries and source type for the COBOL members, even though the new member is added with the updated session settings.

Building member lists

```

IBM Merge Tool for z/OS and OS/390          Row 1 to 7 of 7
Specify Members                               ect option.
                                               Preferences
M                                               -----
-                                               und 2. Batch)
-                                              
"                                              
-                                              
-                                              
-                                              
-                                              
-                                              
*                                              
Current Libraries                               *****
File1: 'USERID.SAZZSAM1'
File2: 'USERID.SAZZSAM2'
Base: 'USERID.SAZZSAMB'
Command ==> _____
Command ==> _____ Scroll ==> PAGE

```

Figure 15. Specify members

2. In the appropriate fields, type the name or name pattern for your File1, File2 and Base Members.
3. If necessary, replace the appropriate Current Libraries fields with the location of these members.
4. Press Enter.

Merge Tool matches the member names based on the pattern supplied and then displays the resulting member list in the main panel.

```

IBM Merge Tool for z/OS and OS/390          Row 1 to 1 of 1
Command ==> _____ Scroll ==> PAGE
Enter "/" to select option.
Member: _____ _ Append to list _ Set Libraries _ Source Type _ Preferences
-----
_ Group Action      _ View Reports    2 Exec Mode ( 1. Foreground 2. Batch)
_ File1            _ File2 Base     Have Have Source
"/" Member Action  Member Member   Work Merge Type
_ AZZCBY01         AZZCBL01 AZZCBL01 YES   COBOL_
***** Bottom of data *****

```

Figure 16. Resulting Member List

Note: The File2 Member and Base Member columns contain the names of the matched members.

Primary Commands for the member list

When building a member list, there are several commands that you can use to manipulate the entire list. These are Primary Commands, and are entered on the Command Line:

Table 6. Member List Primary Commands

Command	Function
CLEAR	Clears the member list area. Once cleared, the member list cannot be restored.
LOCATE or L	Locates a member in the list. Must be followed by a member name/pattern, for example: LOCATE AZZC* OR L AZZSWTCH This command scrolls down the member list to display the first member that matches the pattern provided at the top of the member list area. You can use the UP command to scroll back up the list.
RESET or RES	Clears the "Action" column and restores any Excluded members. See "Using Group Actions" for details about the Member Actions, including Excluded mode.
X [ALL]	Places all members in Excluded mode. Excluded mode can be removed from all members via the RESET command. See "Using Group Actions" for details about Excluded mode.

Using Group Actions

In addition to line command actions that are performed against the individual members, the Merge Tool provides a set of actions that can be performed against the full member list.

The action is performed against all members in the list that have not been excluded via the eXclude line command ("X").

The user can enter the action (GW, GB, GM or SS) at the Group Action field or enter "/" to see group action choices. A group action is applied to all non-excluded members in the current member list.

Building member lists

```

                                IBM Merge Tool for z/OS and OS/390      Row 1 to 5 of 5
                                Merge Tool Group Action                ect option.
M                                Group Action                          Preferences
-                                1. GW-Generate Work File            -----
/                                2. GB-Generate Work and Merge File  und 2. Batch)
"                                3. GM-Generate Merge File from Work File
-                                4. SS-Generate Statistics for Members
-
-                                Select a choice and press ENTER to process group action.
-
-                                Command ==> _____
*
                                *****

Command ==> _____ Scroll ==> PAGE

```

Figure 17. Group Action choices

Include/Exclude actions

The exclude action (X) removes the member from any group action processing performed against the list. The **EXCLUDED** note is placed to the far right of the member that has been excluded. The note remains in this place until the list is rebuilt or the member is the object of the include (I) action.

```

                                IBM Merge Tool for z/OS and OS/390      Row 1 to 9 of 9
                                Enter "/" to select option.
Member: _____ _ Append to list _ Set Libraries _ Source Type _ Preferences
-----
__ Group Action      _ View Reports      1 Exec Mode ( 1. Foreground 2. Batch)
  File1             File2 Base      Have  Have  Source
"/" Member  Action  Member  Member  Work Merge  Type
__ AZZASM01      YES  YES  YES  ASM__
__ AZZCBLA1      YES  YES  YES  COBOL__
__ AZZCBLA2      YES  YES  YES  COBOL__
__ AZZCBL01      YES  YES  YES  COBOL__
__ AZZCBL02      YES  -----  COBOL__
__ AZZCBL11      YES  YES  COBOL__
__ AZZCBY01      -----  YES  COBOL__ EXCLUDED
__ AZZEOFCD      -----  -----  COBOL__ EXCLUDED
__ AZZSWTCH      -----  -----  COBOL__ EXCLUDED
***** Bottom of data *****

Command ==> _____ Scroll ==> PAGE

```

Figure 18. Member Exclude

The Include and Exclude actions can be used in the following ways:

Building member lists

- To exclude individual members, enter the X command in the appropriate Line Command fields.
- To exclude all members in the list, enter the X or X ALL command in the Command line.
- To include individual excluded members, enter the I command in the appropriate Line Command fields.
- To include all excluded members, enter the RESET or RES command in the Command line.

Building member lists

Chapter 3. Analyze complexity and mark-up changes

Once you have identified your files and member names, the next step is to get a picture of the complexity of the changes made and locate overlap areas and conflicts. You can request that Merge Tool generates various statistics and reports that describe these details. Based on the statistics, you can get a better understanding of the resources required for the merge process.

If your reports show that there are overlapping changes and conflicts, you need to mark up those changes in a Work File member and then examine them in detail to resolve the conflicts. A Work File member consists of an intermediate-stage merge of the Base File member with the File1 and File2 insertions, alterations and deletions. These changes are marked with *merge directives* in the first seven bytes of each line. The merge directives determine what happens when the Merge File is built.

If your reports show that there are no overlapping changes or conflicts and you do not want to examine the marked up changes, you can generate the Merge File immediately.

Merge Tool reports and statistics

Merge Tool produces three different types of report outputs. These are:

Table 7. Reports Available

Report Type	Description
Member Statistics	Shows the number of modifications for each source module processed. Conflicting changes are of particular significance.
Summary Statistics	Shows totals related to number of members processed.
Log Messages	Shows the parameters and statements in effect during the report generation, Work File generation or merge process. If processing fails, the reason is reported here.

Generating the reports

You can generate the Statistics reports by running the SS command (Show Statistics for Member) or by generating the Work or Merge Files. A new set of report data is generated each time that you issue the SS command or generate a Work File, Merge File or both. By using the SS command, you can generate the reports without replacing the existing Work or Merge Files. This allows you to edit and save the Work File and then run the reports again, to see if you have resolved all of the conflicts.

To generate the Statistics reports:

1. Ensure that you have provided a data set name for each report that you want to view (see "Specify the data set libraries" on page 11).
2. Enter the SS command as a line command against the appropriate member or as a Group Action. (See "Using Group Actions" on page 25).

Note: If you have not specified data sets for the Member Statistics reports, attempting to use the SS command returns a "Stats DSN needed" error.

Merge Tool reports and statistics

You need to specify the Member Statistics data sets for each member and then regenerate your list before continuing. You do not need to specify Summary Statistics or Log Messages data sets to be able to run the SS command. However, if these data sets are not specified, the resulting report is not generated.

OR

Enter the GW, GM or GB command as a line command against the appropriate member or as a Group Action.

Note: Provided that a report data set was specified at the time that the member was added to the list, the statistical data for the member is generated and included in the reports.

If the SS command was used, the Member Statistics report is displayed when the action is completed.

The statistical data either replaces the current report data sets or is appended to the current reports, depending upon the data set specifications and the *Disp: _ Mod* settings made in the Set Libraries panel. This can have complex implications for the Member Statistics reports - see "Member Statistics reports" on page 31 for details.

Viewing the reports

To view the reports:

1. On the main panel, type a "/" in the View Reports field and press Enter. The View Reports panel is displayed.

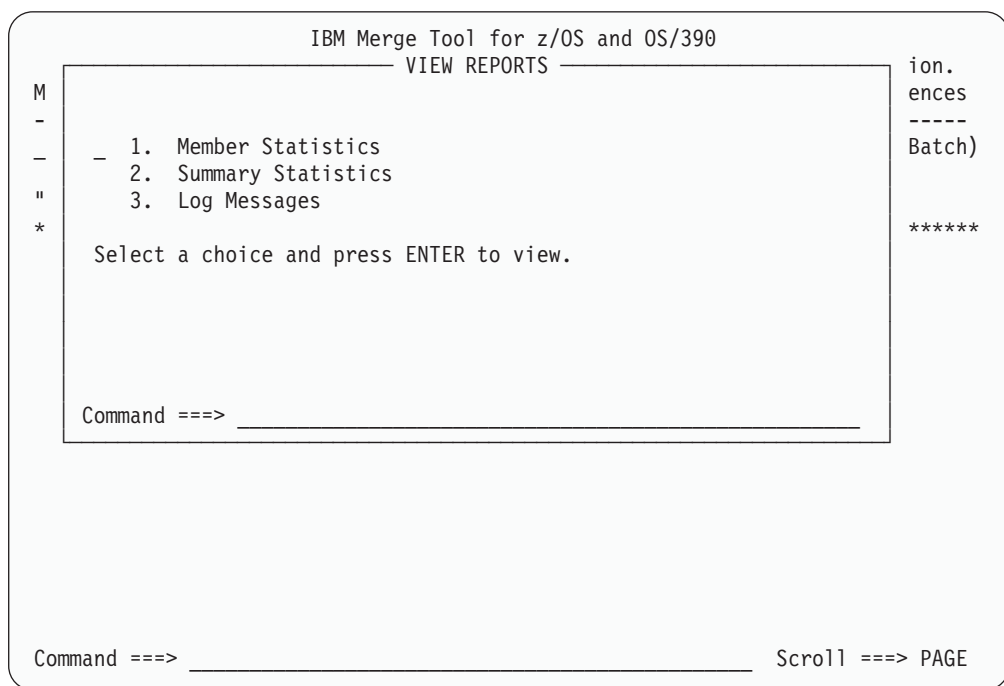


Figure 19. View Reports panel

2. Enter the number of the report you wish to view. The report is displayed in View or Browse mode, (depending upon your Preference settings, see “Displaying the merge components” on page 39).

Notes:

- a. If you have opted to append the report data to existing data sets, you must remember to scroll to the bottom of the information when you view the reports.
- b. If you have used different data sets for various members on the list, you need to change data set name in the Set Libraries panel to view the reports. The View Reports command reads the active data set names on that panel.
- c. If you have not allocated a data set for the selected report, a message is displayed. You must specify, in the Set Libraries panel, the data set that you want to view.

Report structure

Member Statistics reports

A Member Statistics report is generated each time you issue a mark-up request command. The structure of this report differs, depending upon the Set Library options used when each member was added to the list and the way in which the Work File was generated.

The simplest scenario available is where a single member has been added to the list, with the Disp: mod field cleared. When a mark-up request is performed using the Line Command field for that member, the report data is written to the specified data set, replacing any existing information.

```
***** Top of Data *****
16:01:11 ----- WORK FILE STATISTICS -----
16:01:11 File1 name..... USERID.SAZZSAM1(AZZCBLA1)
16:01:11 File2 name..... USERID.SAZZSAM2(AZZCBLA1)
16:01:11 Baseline name..... USERID.SAZZSAMB(AZZCBLA1)
16:01:11 # of lines in base.... 96
16:01:11 # of lines in file1... 117
16:01:11 # of lines in file2... 101
16:01:11 # of unchanged lines.. 85
16:01:11 # of inserts both.... 0
16:01:11 # of inserts file1.... 27
16:01:11 # of inserts file2.... 12
16:01:11 # of deletes both.... 2
16:01:11 # of deletes file1.... 4
16:01:11 # of deletes file2.... 5
16:01:11 # of conflicts both... 2
16:01:11 # of conflicts file1.. 2
16:01:11 # of conflicts file2.. 3
16:01:11 -----
***** Bottom of Data *****
```

Figure 20. Member Statistics report for single member Mark-up Request

Note: While most of these statistics are self-explanatory, the “# of conflicts” lines merit further clarification. In this example, the report shows that 2 lines are in conflict for “both” and for “file1” but 3 lines are in conflict for “file2”. This has occurred because, in one location, File1 and File2 deleted the same line and replaced it with a different new line, causing a conflict. However, in File2 the next line was also deleted. MergeTool does not ordinarily flag deletions as a conflict (it was the conflicting replacement lines that caused

Merge Tool reports and statistics

the flag), however, it marks *any* activity that occurs around an existing marked conflict as a conflict. Therefore, this particular situation is recorded as 1 conflict in the “file1” and “both” counts and 2 conflicts in “file2” count.

In this scenario, if a second markup request was sent with the Disp: mod field selected, the new data would be appended to the existing information.

```
***** Top of Data *****
16:01:11 ----- WORK FILE STATISTICS -----
16:01:11 File1 name..... USERID.SAZZSAM1(AZZCBLA1)
16:01:11 File2 name..... USERID.SAZZSAM2(AZZCBLA1)
16:01:11 Baseline name..... USERID.SAZZSAMB(AZZCBLA1)
16:01:11 # of lines in base... 96
16:01:11 # of lines in file1... 117
16:01:11 # of lines in file2... 101
16:01:11 # of unchanged lines.. 85
16:01:11 # of inserts both.... 0
16:01:11 # of inserts file1... 27
16:01:11 # of inserts file2... 12
16:01:11 # of deletes both.... 2
16:01:11 # of deletes file1... 4
16:01:11 # of deletes file2... 5
16:01:11 # of conflicts both... 2
16:01:11 # of conflicts file1.. 2
16:01:11 # of conflicts file2.. 3
16:01:11 -----
11:22:10 ----- WORK FILE STATISTICS -----
11:22:10 File1 name..... USERID.SAZZSAM1(AZZCBLA1)
11:22:10 File2 name..... USERID.SAZZSAM2(AZZCBLA1)
11:22:10 Baseline name..... USERID.SAZZSAMB(AZZCBLA1)
11:22:10 # of lines in base... 96
11:22:10 # of lines in file1... 117
11:22:10 # of lines in file2... 101
11:22:10 # of unchanged lines.. 85
11:22:10 # of inserts both.... 0
11:22:10 # of inserts file1... 27
11:22:10 # of inserts file2... 12
11:22:10 # of deletes both.... 2
11:22:10 # of deletes file1... 4
11:22:10 # of deletes file2... 5
11:22:10 # of conflicts both... 2
11:22:10 # of conflicts file1.. 2
11:22:10 # of conflicts file2.. 3
11:22:10 -----
***** Bottom of Data *****
```

Figure 21. Member Statistics report for single member with appended data

When multiple members with the same Report data set locations have been added to the list, using Line Commands or Group Actions to generate the Work File has differing results, depending upon the Disp: mod setting.

- **Disp: mod is cleared and Group Action used:** The resulting file displays the Member Statistics for each non-excluded list member, replacing any previous data.
- **Disp: mod is selected and Group Action used:** The resulting file displays the Member Statistics for each non-excluded list member, appending to any previous data.
- **Disp: mod is cleared and multiple Line Commands are used:** The resulting file displays only the Member Statistics for the last non-excluded list member given a mark-up request, replacing any previous data.

- **Disp: mod is selected and multiple Line Commands are used:** The resulting file displays the Member Statistics for each non-excluded list member given a mark-up request, appending to any previous data.

The most complex scenario for Member Statistics, is where the list members have been appended with different Report data set locations and different Disp: mod settings. In this case, Merge Tool is able to send each member to the appropriate Report data set, appending or replacing the existing data according to the individual Disp: mod settings.

Note: Do not mix the Disp: mod settings for individual members where the Report is being sent to the same data set. This might cause unexpected results. For example, if three members were added to the list as follows:

- AZZCBLA1: Disp: mod cleared (replace).
- AZZCBLA2: Disp: mod selected (append).
- AZZCBL01: Disp: mod cleared (replace).

and then a mark-up request was entered against each member, Merge Tool processes each request in turn and the resulting report would show AZZCBL01 only.

Summary Statistics report

The Summary Statistics report shows the number of matching members existing in the File1, File2 and Base data sets.

When you generate files using Foreground Execution, only members of File2 and Base that match members of File1 appear in the report matrix. Additionally, any File1 members that do not have a match to File2 and Base members appear in the generated report. In this case, the “# of files existing in F2 and Baseline only”, “# of files existing in F2 only”, “# of files existing in Baseline only”, and “# of files not found in any library” values are always 0.

```
***** Top of Data *****
15:26:18 ----- SUMMARY STATISTICS -----
15:26:18 # of files existing in F1, F2, and Baseline.. 4
15:26:18 # of files existing in F1 and F2 only..... 1
15:26:18 # of files existing in F1 only..... 2
15:26:18 # of files existing in F1 and Baseline only.. 1
15:26:18 # of files existing in F2 and Baseline only.. 0
15:26:18 # of files existing in F2 only..... 0
15:26:18 # of files existing in Baseline only..... 0
15:26:18 # of files not found in any library..... 0
15:26:18 -----
***** Bottom of Data *****
```

Figure 22. Summary Statistics report - Foreground generation

When you generate files using Batch Execution, and use wild cards to specify the members, MergeTool looks through all of the specified File1, File2 and Base libraries. In this way it is able to report on the “# of files existing in F2 and Baseline only”, the “# of files existing in F2 only” and the “# of files existing in Baseline only”.

Merge Tool reports and statistics

```
***** Top of Data *****
15:43:06 ----- SUMMARY STATISTICS -----
15:43:06 # of files existing in F1, F2, and Baseline.. 4
15:43:06 # of files existing in F1 and F2 only..... 1
15:43:06 # of files existing in F1 only..... 2
15:43:06 # of files existing in F1 and Baseline only.. 1
15:43:06 # of files existing in F2 and Baseline only.. 2
15:43:06 # of files existing in F2 only..... 0
15:43:06 # of files existing in Baseline only..... 1
15:43:06 # of files not found in any library..... 0
15:43:06 -----
***** Bottom of Data *****
```

Figure 23. Summary Statistics report - Batch generation with wild cards

When you generate files using Batch Execution and specify the members by name (no wild cards), any files that are not found are included in the “# of files not found in any library” count.

```
***** Top of Data *****
15:45:01 ----- SUMMARY STATISTICS -----
15:45:01 # of files existing in F1, F2, and Baseline.. 0
15:45:01 # of files existing in F1 and F2 only..... 0
15:45:01 # of files existing in F1 only..... 0
15:45:01 # of files existing in F1 and Baseline only.. 0
15:45:01 # of files existing in F2 and Baseline only.. 0
15:45:01 # of files existing in F2 only..... 0
15:45:01 # of files existing in Baseline only..... 0
15:45:01 # of files not found in any library..... 1
15:45:01 -----
***** Bottom of Data *****
```

Figure 24. Summary Statistics report - Batch generation with named member

Log Messages report

The Log Messages report shows the actions that took place when your files were last generated. This information includes:

- the time at which a file was written;
- the type of file generated (Work File, Merge File, Summary Statistics or Member Statistics);
- the data set to which the file was written; and
- the number of lines written to each file (for Work and Merge Files only).

In the example shown below, the GW command was applied as a group action to the sample files. For each member added to the MergeTool list, the same data set was specified as the Work File, Summary Statistics and Member Statistics output locations.

```

***** Top of Data *****
13:42:52 -----
13:42:52 - IBM MERGE TOOL FOR z/OS AND OS/390
13:42:52 - Log file - 12 Jun 2003
13:42:52 -----
13:43:07 Work file written: USERID.SAZZSAMW(AZZASM01)
13:43:07 Lines written....: 90
13:43:12 Work file written: USERID.SAZZSAMW(AZZCBLA1)
13:43:12 Lines written....: 141
13:43:16 Work file written: USERID.SAZZSAMW(AZZCBLA2)
13:43:16 Lines written....: 106
13:43:21 Work file written: USERID.SAZZSAMW(AZZCBL01)
13:43:21 Lines written....: 111
13:43:26 Work file written: USERID.SAZZSAMW(AZZCBL02)
13:43:26 Lines written....: 137
13:43:31 Work file written: USERID.SAZZSAMW(AZZCBL01)
13:43:31 Lines written....: 103
13:43:36 Work file written: USERID.SAZZSAMW(AZZEFC01)
13:43:36 Lines written....: 8
13:43:41 Work file written: USERID.SAZZSAMW(AZZSWTCH)
13:43:41 Lines written....: 9
13:43:42 Summary Statistics Written to USERID.SAZZSAMS
13:43:42 Member Statistics Written to USERID.SAZZSAMZ
***** Bottom of Data *****

```

Figure 25. Log Messages report - simple output

In the following example, the AZZASM01 member was added to the list with all libraries, except the Log Messages file, set to ASM.SAZZSAM*. The AZZCB* members were added to the list with the libraries set to COBOL.SAZZSAM*. For all of these members, the Log Messages output was set to SAZZSAML. The GB command was then applied as a group action.

The report shows that the ASM member and its Summary and Member Statistics reports, were correctly written to the ASM.SAZZSAM* locations; while the COBOL members and reports were written to COBOL.SAZZSAM*.

Marking up the changes

```
***** Top of Data *****
14:40:13 -----
14:40:13 - IBM MERGE TOOL FOR z/OS AND OS/390
14:40:13 - Log file - 12 Jun 2003
14:40:13 -----
14:40:20 Work file written: USERID.ASM.SAZZSAMW(AZZASM01)
14:40:20 Lines written....: 87
14:40:22 Merge file written: USERID.ASM.SAZZSAMM(AZZASM01)
14:40:22 Lines written....: 62
14:40:22 Summary Statistics Written to USERID.ASM.SAZZSAMS
14:40:22 Member Statistics Written to USERID.ASM.SAZZSAMZ
14:40:31 -----
14:40:31 - IBM MERGE TOOL FOR z/OS AND OS/390
14:40:31 - Log file - 12 Jun 2003
14:40:31 -----
14:40:43 Work file written: USERID.COBOL.SAZZSAMW(AZZCBLA1)
14:40:43 Lines written....: 141
14:40:44 Merge file written: USERID.COBOL.SAZZSAMM(AZZCBLA1)
14:40:44 Lines written....: 124
14:40:48 Work file written: USERID.COBOL.SAZZSAMW(AZZCBLA2)
14:40:48 Lines written....: 106
14:40:49 Merge file written: USERID.COBOL.SAZZSAMM(AZZCBLA2)
14:40:49 Lines written....: 92
14:40:53 Work file written: USERID.COBOL.SAZZSAMW(AZZCBL01)
14:40:53 Lines written....: 111
14:40:55 Merge file written: USERID.COBOL.SAZZSAMM(AZZCBL01)
14:40:55 Lines written....: 97
14:40:58 Work file written: USERID.COBOL.SAZZSAMW(AZZCBL02)
14:40:58 Lines written....: 137
14:41:00 Merge file written: USERID.COBOL.SAZZSAMM(AZZCBL02)
14:41:00 Lines written....: 97
14:41:03 Work file written: USERID.COBOL.SAZZSAMW(AZZCBL01)
14:41:03 Lines written....: 103
14:41:05 Merge file written: USERID.COBOL.SAZZSAMM(AZZCBL01)
14:41:05 Lines written....: 97
14:41:05 Summary Statistics Written to USERID.COBOL.SAZZSAMS
14:41:05 Member Statistics Written to USERID.COBOL.SAZZSAMZ
***** Bottom of Data *****
```

Figure 26. Log Messages report - complex output

Marking up the changes

Work Files and reports can be generated for individual list members, using Line Commands, or for all members in the list, using Group Actions. The file generation can run in the Foreground or as a Batch Job, depending on the value in the Exec Mode field. This section describes the Foreground execution process. For a description of Batch mode execution, see Chapter 6, “The Merge Tool batch utility,” on page 55.

Note: When using Group Actions, you can exclude selected members from the list. See “Using Group Actions” on page 25.

There are two variations of the command that generates a Work File. These are:

GW Generate Work File.

This command creates the Work File from the Base, File1 and File2 members. Once generated, you are able to examine the file and, if necessary, modify it to resolve conflicts. If you have specified a data set for each of your Report types, it also generates that report.

GB Generate Work and Merge File.

This command creates both the Work and Merge Files in one action. As there is no opportunity to modify the Work File during the process, it is most commonly used when you have already examined a Work File and have determined that there are no conflicts or when you have resolved all conflicts by modifying the File1 or File2 members, not the Work File member.

Note: If you have previously run a GW command and then modified the Work File, running this command replaces the existing (modified) Work File.

These commands are collectively known as *Mark-up Request* commands.

Generating the Work File

To generate a Work File:

1. Build your list of members, using the techniques described in Chapter 2, "Identify the components to be examined," on page 11.

Note: If you want to view the statistics reports, you must have added your members to the list with an allocated data set library for each desired report type. (See "Specify the data set libraries" on page 11 for details.)

2. Set the Execution Mode field to 1. Foreground. For information about using Batch execution, see Chapter 6, "The Merge Tool batch utility," on page 55.
3. Type the GW or GB command in the Line Command field adjacent to the relevant list member.

OR

Type the GW or GB command in the Group Action field. This generates a Work File for all included members.

OR

Type a "/" into the Line Command or Group Action field. Select GW or GB from the pop-up list of commands.

4. Press Enter. The Work File generation begins and a Progress Status message is displayed in a pop-up window.

Marking up the changes

```

                                IBM Merge Tool for z/OS and OS/390          Row 1 to 5 of 5
                                Progress Indicator                          "/" to select option.
M                                Program : AZZCBLA1                      urce Type _ Preferences
-                                Request : Build work file                -----
-                                Function: Analyzing request              1. Foreground 2. Batch)
"                                Source
                                Type
                                COBOL__
GW AZZCBLA1                      YES    YES    YES    COBOL__
  AZZCBLA2                      YES    YES    YES    COBOL__
  AZZCBL01                      YES    YES    YES    COBOL__
  AZZCBL02                      YES    ----- COBOL__
***** Bottom of data *****
Command ==> _____ Scroll ==> PAGE

```

Figure 27. Progress Status Message

- When the process is completed, each selected member contains "YES" in the Have Work column and "PROCESSED" in the Action column.

The Action column always displays the last action performed on the member. If you are performing multiple actions on selected members, you might want to clear this column from time to time, so that you can more easily determine which members were affected by your most recent action.

To clear the Action column, enter RESET or RES on the Command Line.

Chapter 4. Review changes and address conflicts

Upon successful creation of a Work File member, you must determine if there are any conflicting changes between the member in File1 and the member in File2.

Any conflicting changes must be reviewed. If you determine that there is a conflict in overlapping changes, you must address these conflicts prior to merging the source into a combined member. Conflicting changes are indicated in the statistics reports as well as in the Work File members.

The overall process needed to review changes and address conflicts is to:

1. Display the merge components, particularly the Work File, in View or Browse mode.
2. Interpret the mark-up indicators and locate the source of conflicts.
3. Edit the Work File, so that the lines no longer conflict

OR

Edit the File1 or File2 member, so that the lines no longer conflict, and then regenerate the Work File.

Displaying the merge components

While interpreting the mark-up indicators in the Work File, you might also want to display the Base, File1 or File2 members. This could aid you in determining the best location in which to make any changes needed to resolve a conflict. If the Merge File has also been generated, you can use this member to examine the consequences of merging the unchanged Work File. This might help you in your decisions.

The merge component members can be displayed in an ISPF Browse or View session. By default, Merge Tool uses Browse mode to display members. If you prefer to use View mode, you must change your Merge Tool Preferences.

To specify View or Browse mode:

1. On the main panel, enter a "/" in the Preferences field. The Merge Tool Option Default Settings panel is displayed.

Displaying the merge components

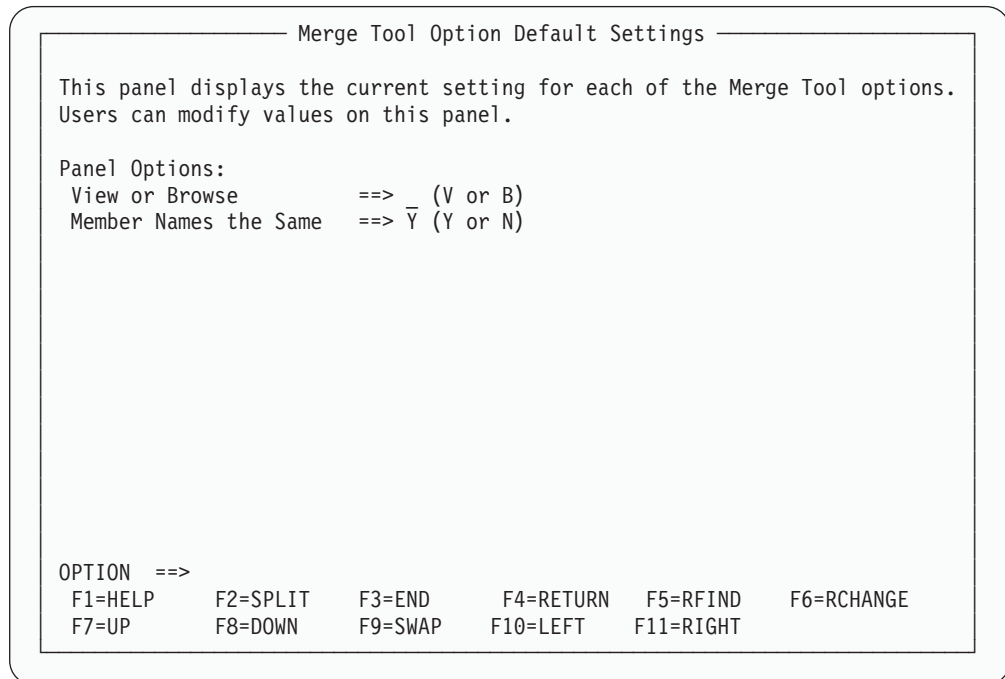


Figure 28. The Merge Tool Option Default Settings panel (Preferences)

2. Enter V (View) or B (Browse) in the View or Browse field.

Note: If the View or Browse field is cleared, the default setting of Browse is used.

There are several ways to display the merge component members. You can enter any one of the following commands in the Line Command field for the member to be reviewed:

BB, B1, B2, BW or BM

Browse Base, File1, File2, Work File or Merge File. These commands display the selected member immediately.

- B** Browse. This command displays a list of the merge component members.

Displaying the merge components

```

IBM Merge Tool for z/OS and OS/390      Row 1 to 8 of 8
Merge Tool Member Actions                ect option.
                                         Preferences
                                         -----
                                         und 2. Batch)

M
-   Select desired library: _   Member: AZZCBLA2
-
-   1. File1 'USERID.SAZZSAM1'
"   2. File2 'USERID.SAZZSAM2'
-   3. Base  'USERID.SAZZSAMB'
-   4. Work  'USERID.SAZZSAMW'
B   5. Merge 'USERID.SAZZSAMB'

-   Select library and press ENTER to BROWSE
-
-   Command ==>
-   F1=HELP   F2=SPLIT   F3=END   F4=RETURN   F5=RFIND
*   F6=RCHANGE F7=UP     F8=DOWN  F9=SWAP   F10=LEFT   *****

Command ==>                               Scroll ==> PAGE
F1=HELP   F2=SPLIT   F3=END   F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=RETRIEVE

```

Figure 29.

Enter the number for the required member in the Select desired library field and press Enter.

/ This command displays the list of available Member Actions.

```

IBM Merge Tool for z/OS and OS/390      Row 1 to 8 of 8
Merge Tool Member Actions                ect option.
                                         Preferences
                                         -----
                                         und 2. Batch)

M
-   Member Action for member AZZCBLA2
-
-   1. GW-Generate Work File
"   2. GB-Generate Work and Merge File
-   3. GM-Generate Merge File from Work File
-   4. SS-Show Statistics for Member
/   5. E (EB, E1, E2, EW, EM) - Edit Member
-   6. B (BB, B1, B2, BW, BM) - Browse Member
-   7. X - Exclude Member
-   8. I - Include Excluded Member
-   Command ==>
-   F1=HELP   F2=SPLIT   F3=END   F4=RETURN   F5=RFIND
*   F6=RCHANGE F7=UP     F8=DOWN  F9=SWAP   F10=LEFT   *****

Command ==>                               Scroll ==> PAGE
F1=HELP   F2=SPLIT   F3=END   F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=RETRIEVE

```

Figure 30. Member Actions

Enter BB, B1, B2, BW or BM and press Enter to display the required merge component immediately.

OR

Displaying the merge components

Enter 6 or B to display the list of merge components (see 40), and then select from that list.

Note: "B" indicates "Browse or View". The command does not change when your Preference is set to View mode.

Interpreting the Work File

The marked-up file shows how modifications in the first version of the source module are merged with modifications in the second version. Columns one through five of the marked-up file indicate whether or not a line was modified.

- Column one indicates if a line was modified. Modified lines are shown with a "." in column one.
- Column two indicates conflicting lines. Conflicting lines are shown with a "?" in column two.
- Column three indicates how the line was modified. Inserts are shown with a "+" while deletes are shown with a "-" in column three. A line alteration is shown as an insertion of a new line (with a "+" in column 3), followed by a deletion of the original line (with a "-" in column 3).
- Column four and five indicate the origin of the modification (##): "1" indicating File1, "2" indicating File2, and "12" indicating that both files contain the same modification.

The following sections show examples of the mark-up indicators that appear in the Work File when changes are made in File1 and File2, and the resulting consequences in the output merged file. The sample member AZZCLBA1 is used in all examples.

Line insertion example

This example shows the mark-up indicators that appear when lines are inserted into File1 and File2 without causing a conflict.

The Base member (original version) contains:

```
000093      ELSE
000094      WRITE REC-REC-OUT FROM REC-REC-IN
000095      END-IF.
000096      READ REC-IN AT END SET TRUE-END-OF-FILE TO TRUE.
***** ***** Bottom of Data *****
```

Figure 31. Line insertion example: Base member

Lines in bold have been inserted in the File1 member but not in the File2 member.

```
000111      ELSE
000112      WRITE REC-REC-OUT FROM REC-REC-IN
000113      END-IF.
000114      READ REC-IN AT END SET TRUE-END-OF-FILE TO TRUE.
000115      130-OBTAIN-NAME
000116      READ NAME-IN AT END SET TRUE-GOT-NAME TO TRUE.
000117      MOVE LAST TO LAST-NAME.
***** ***** Bottom of Data *****
```

Figure 32. Line insertion example: File1 member

Note: The line numbers do not match the Base member as other insertions have taken place above this point.

Lines in bold have been inserted in the File2 member but not in the File1 member.

```
000094         ELSE
000095         WRITE REC-REC-OUT FROM REC-REC-IN
000096         END-IF.
000097         READ REC-IN AT END SET TRUE-END-OF-FILE TO TRUE.
000098
000099         999-CLOSE-OUT.
000100         CLOSE REC-IN, REC-OUT, REC-REP.
000101         STOP RUN.
***** ***** Bottom of Data *****
```

Figure 33. Line insertion example: File2 member

The Work File displays the intermediate merge stage with mark-up indicators. The indicators show that:

- lines 000134 to 000140 have been modified ("." in column 1)
- lines 000134 to 000136 were added in File1 (+1 in columns 3 and 4)
- lines 000137 to 000140 were added in File2 (+2 in columns 3 and 4)

```
000130         ELSE
000131         WRITE REC-REC-OUT FROM REC-REC-IN
000132         END-IF.
000133         READ REC-IN AT END SET TRUE-END-OF-FILE TO TRUE.
000134 . +1         130-OBTAIN-NAME
000135 . +1         READ NAME-IN AT END SET TRUE-GOT-NAME TO TRUE.
000136 . +1         MOVE LAST TO LAST-NAME.
000137 . +2
000138 . +2         999-CLOSE-OUT.
000139 . +2         CLOSE REC-IN, REC-OUT, REC-REP.
000140 . +2         STOP RUN.
***** ***** Bottom of Data *****
```

Figure 34. Line insertion example: Work File member

The final Merge File includes the lines from both File1 and File2.

```
000114         ELSE
000115         WRITE REC-REC-OUT FROM REC-REC-IN
000116         END-IF.
000117         READ REC-IN AT END SET TRUE-END-OF-FILE TO TRUE.
000118         130-OBTAIN-NAME
000119         READ NAME-IN AT END SET TRUE-GOT-NAME TO TRUE.
000120         MOVE LAST TO LAST-NAME.
000121
000122         999-CLOSE-OUT.
000123         CLOSE REC-IN, REC-OUT, REC-REP.
000124         STOP RUN.
***** ***** Bottom of Data *****
```

Figure 35. Line insertion example: Merge File member

Line deletion example

This example shows the mark-up indicators that appear when lines are deleted from File1 and File2 without causing a conflict.

Interpreting the Work File

Note: Carefully examine all deletions that occur in one modified file only, for possible consequences in the other file. Because the matching line has not been removed from the second file, there is a possibility that the code in the second file relies on statements made in the "deleted" line.

One situation in which this might occur is when the person coding File1 adds new statements that use information supplied in the deleted line. He is unaware that the person coding File2 has deleted the line. The person coding File2 can resolve any existing problems caused by removing the line, but he is unaware that the File1 coder has added new statements. When these files are merged, the deletion might cause the new code to fail because the statements added to File1 no longer work when the line removed in File2 has been deleted. Such deletions are displayed in the Work File but are not indicated as a conflict, as the File1 coder has not changed the line.

The Base member (original version) contains:

```
000079      120-FIND-PROCESS-NAMES.  
000080  
000081      MOVE 0 TO RES-NAME-NAME.
```

Figure 36. Line deletion example: Base member

These lines have been retained in the File1 member.

```
000096      120-FIND-PROCESS-NAMES.  
000097  
000098      MOVE 0 TO RES-NAME-NAME.
```

Figure 37. Line deletion example: File1 member

The blank line between the statements has been deleted in the File2 member but not in the File1 member.

```
000081      120-FIND-PROCESS-NAMES.  
000082      MOVE 0 TO RES-NAME-NAME.
```

Figure 38. Line deletion example: File2 member

The Work File displays the intermediate merge stage with mark-up indicators. The indicators show that:

- line 000113 was modified ("." in column 1)
- the line was deleted in File2 ("-2" in columns 3 and 4)

```
000112      120-FIND-PROCESS-NAMES.  
000113 . -2  
000114      MOVE 0 TO RES-NAME-NAME.
```

Figure 39. Line deletion example: Work File member

The final Merge File shows that the line has been removed. While this simple deletion is unlikely to cause problems, if a more complex line were to be deleted in File2, there could be consequences in the merged file.

```
000100      120-FIND-PROCESS-NAMES.
000101      MOVE 0 TO RES-NAME-NAME.
```

Figure 40. Line deletion example: Merge File member

Line alteration example

This example shows the mark-up indicators that appear when lines are altered in File1 and File2 without causing a conflict.

Note: Alterations are shown as an insertion and deletion for each line changed.

The Base member (original version) contains:

```
000057      01 FULL-NAME-REC          PIC X(80)      VALUE SPACES.
000058      01 SF-STAT                PIC X(2)      .
```

Figure 41. Line alteration example: Base member

The first of these lines has been changed in the File1 member.

```
000073      01 LAST-NAME            PIC X(30)      VALUE SPACES.
000074      01 SF-STAT                PIC X(2)      .
```

Figure 42. Line alteration example: File1 member

The second of these lines has been changed in the File2 member.

```
000057      01 FULL-NAME-REC          PIC X(80)      VALUE SPACES.
000058      01 SF-STAT                PIC X(2)      VALUE SPACES.
```

Figure 43. Line alteration example: File2 member

The Work File displays the intermediate merge stage with mark-up indicators. The indicators show that:

- lines 000080 to 000083 contain modifications (". " in column 1)
- lines 000080 and 000081 were the result of an alteration in File1 (" +1 " and " -1 " in columns 3 and 4)
- lines 000082 and 000083 were the result of an alteration in File2 (" +2 " and " -2 " in columns 3 and 4)

```
000080 . +1      01 LAST-NAME            PIC X(30)      VALUE SPACES.
000081 . -1      01 FULL-NAME-REC          PIC X(80)      VALUE SPACES.
000082 . +2      01 SF-STAT                PIC X(2)      VALUE SPACES.
000083 . -2      01 SF-STAT                PIC X(2)      .
```

Figure 44. Line alteration example: Work File member

The final Merge File shows the two altered lines.

Interpreting the Work File

```
000074      01  LAST-NAME          PIC X(30)      VALUE SPACES.  
000075      01  SF-STAT           PIC X(2)       VALUE SPACES.
```

Figure 45. Line alteration example: Merge File member

Conflicting source changes example

This simple example of conflicting source changes shows the mark-up indicators that appear when both File1 and File2 attempt to modify the same line.

The Base member (original version) contains:

```
000001      IDENTIFICATION DIVISION.  
000002      PROGRAM-ID.      AZZCBLA1.  
000003      AUTHOR.         ARTHUR AUTHOR.  
000004      *****
```

Figure 46. Conflicting alterations example: Base member

The Author's name has been changed to "Miller" in the File1 member.

```
000001      IDENTIFICATION DIVISION.  
000002      PROGRAM-ID.      AZZCBLA1.  
000003      AUTHOR.         ARTHUR MILLER.  
000004      *****
```

Figure 47. Conflicting alterations example: File1 member

In the File2 member, the Author's name is changed to "Murray".

```
000001      IDENTIFICATION DIVISION.  
000002      PROGRAM-ID.      AZZCBLA1.  
000003      AUTHOR.         ARTHUR MURRAY.  
000004      *****
```

Figure 48. Conflicting alterations example: File2 member

The Work File displays the intermediate merge stage with mark-up indicators. The indicators show that:

- lines 000008 to 000010 contain modifications which are causing a conflict ("." in column 1 and "?" in column 2)
- line 000008 was inserted in File1 (" +1" in columns 3 and 4)
- line 000009 was inserted in File2 (" +2" in columns 3 and 4)
- line 000010 was deleted in both File1 and File2 (" -12" in columns 3, 4 and 5)

```
000006      IDENTIFICATION DIVISION.  
000007      PROGRAM-ID.      AZZCBLA1.  
000008 .?+1      AUTHOR.         ARTHUR MILLER.  
000009 .?+2      AUTHOR.         ARTHUR MURRAY.  
000010 .?-12     AUTHOR.         ARTHUR AUTHOR.  
000011      *****
```

Figure 49. Conflicting alterations example: Work File member

Note:

If the Work File were allowed to merge without resolving this conflict, the final Merge File shows both inserted lines.

```
000001      IDENTIFICATION DIVISION.
000002      PROGRAM-ID.      AZZCBLA1.
000003      AUTHOR.          ARTHUR MILLER.
000004      AUTHOR.          ARTHUR MURRAY.
000005      *****
```

Figure 50. Conflicting alterations example: Merge File member

To resolve this conflict, the reviewer can edit the Work File or edit the source files. In this case the simplest resolution would be to decide upon the correct name and then, in the Work File, delete the mark-up indicators from that line, then delete the other two lines:

```
000006      IDENTIFICATION DIVISION.
000007      PROGRAM-ID.      AZZCBLA1.
000008      AUTHOR.          ARTHUR MILLER.
000009      *****
```

Figure 51. Conflicting alterations example: edited Work File member

A more complex situation is shown in the following example:

The Base member (original version) contains:

```
000069      000-MAIN.
000070
000071      OPEN INPUT  REC-IN
000072      OUTPUT REC-OUT, REC-REP.
000073      SET FALSE-END-OF-FILE TO TRUE.
```

Figure 52. Complex conflicts example: Base member

The File1 coder, for whatever reason, needs to open another file definition at this point, so he has added an FD statement earlier in the code and alters this line to open the file.

Interpreting the Work File

```
000036      FD  NAME-IN
000037          LABEL RECORDS ARE STANDARD
000038          RECORD CONTAINS 80 CHARACTERS
000039          BLOCK CONTAINS 0 RECORDS
000040          RECORDING MODE IS F.
000041      01  NAME-REC-IN.
000042          05  LAST                PIC X(30).
000043          05  FIRST               PIC X(15).
000044          05  MIDDLE              PIC X(15).
000045          05  PREFIX              PIC X(3).
000046          05  FILLER             PIC X(37).
          :
000086      000-MAIN.
000087
000088          OPEN INPUT  REC-IN, NAME-IN
000089                  OUTPUT REC-OUT, REC-REP.
000090          SET FALSE-END-OF-FILE TO TRUE.
```

Figure 53. Complex conflicts example: File1 member

At the same time, the File2 coder has decided to turn the OPEN statement into a separate procedure, which he might perform several times. He inserts the new procedure after MAIN and then changes the OPEN statement to perform a call to this new code.

```
000069      000-MAIN.
000070
000071          PERFORM 110-OPEN-FILES.
000072          SET FALSE-END-OF-FILE TO TRUE.
          :
000077      110-OPEN-FILES.
000078          OPEN INPUT  REC-IN
000079                  OUTPUT REC-OUT, REC-REP.
```

Figure 54. Complex conflicts example: File2 member

The Work File shows both the new FD statement and the new procedure as simple insertions (lines 000043 to 000053 and 000106 to 000109) but finds a conflict in the MAIN procedure (lines 000097 to 000100).


```

000043 . +1      FD  NAME-IN
000044 . +1      LABEL RECORDS ARE STANDARD
000045 . +1      RECORD CONTAINS 80 CHARACTERS
000046 . +1      BLOCK CONTAINS 0 RECORDS
000047 . +1      RECORDING MODE IS F.
000048 . +1      01  NAME-REC-IN.
000049 . +1      05  LAST                PIC X(30).
000050 . +1      05  FIRST                PIC X(15).
000051 . +1      05  MIDDLE               PIC X(15).
000052 . +1      05  PREFIX               PIC X(3).
000053 . +1      05  FILLER               PIC X(37).
:
:
000095          000-MAIN.
000096
000097 .?+1      OPEN INPUT REC-IN, NAME-IN
000098 .?+2      PERFORM 110-OPEN-FILES.
000099 .?-12     OPEN INPUT REC-IN
000100 .?-2      OUTPUT REC-OUT, REC-REP.
000101          SET FALSE-END-OF-FILE TO TRUE.
:
:
000106 . +2      110-OPEN-FILES.
000107 . +2      OPEN INPUT REC-IN
000108 . +2      OUTPUT REC-OUT, REC-REP.

```

Figure 55. Complex conflicts example: Work File member

If the Work File were allowed to merge without resolving this conflict, the final Merge File causes run-time problems as it attempts to open REC-IN twice.

```

000087          000-MAIN.
000088
000089          OPEN INPUT REC-IN, NAME-IN
000090          PERFORM 110-OPEN-FILES.
000091          SET FALSE-END-OF-FILE TO TRUE.

```

Figure 56. Complex conflicts example: Merge File member

One possible solution to this would be to alter the Work File so that NAME-IN is added to the 110-OPEN-FILES routine and the repeating line is removed.

```

000095          000-MAIN.
000096
000097 .?+2      PERFORM 110-OPEN-FILES.
000098 .?-12     OPEN INPUT REC-IN
000099 .?-12     OUTPUT REC-OUT, REC-REP.
000100          SET FALSE-END-OF-FILE TO TRUE.
:
:
000105 . +2      110-OPEN-FILES.
000106 . +2      OPEN INPUT REC-IN, NAME-IN
000107 . +2      OUTPUT REC-OUT, REC-REP.

```

Figure 57. Complex conflicts example: edited Work File member

Note: In this file, the following changes have been made:

- Line 000097 was deleted (remaining lines are renumbered).
- The mark-up indicators in line 000100 (now 000099) were changed from ".?-2" to ".?-12".
- NAME-IN was added to line 000106.

Interpreting the Work File

The Work File can then be used to generate a new Merge File.

```
000086      000-MAIN.  
000087  
000088      PERFORM 110-OPEN-FILES.  
000089      SET FALSE-END-OF-FILE TO TRUE.  
:  
:  
000094      110-OPEN-FILES.  
000095      OPEN INPUT REC-IN, NAME-IN  
000096      OUTPUT REC-OUT, REC-REP.
```

Figure 58. Complex conflicts example: resolved Merge File member

Editing the Work File or source files

There are several ways to edit the merge component members. You can enter any one of the following commands in the Line Command field for the member to be opened in Edit mode:

EB, E1, E2, EW or EM

Edit Base, File1, File2, Work File or Merge File. These commands display the selected member immediately.

E Edit. This command displays a list of the merge component members (see Figure 29 on page 41). Enter the number for the required member in the Select desired library field and press Enter.

/ This command displays the list of available Member Actions (see Figure 30 on page 41).

Enter EB, E1, E2, EW or EM and press Enter to display the required merge component immediately.

OR

Enter 5 or E to display the list of merge components, and then select from that list.

Notes:

1. If you edit a source member (Base, File1 or File2) that is under SCLM control, you erase the SCLM accounting information. To restore this information, you need to reopen the source member using SCLM Edit and then save the member.
2. If you edit the Work File and then use the GW or GB command again, you erase any changes you have made. Once the Work File has been edited, use the GM command when you want to generate a new Merge File.

Chapter 5. Creating the merged source

Once the Work File has been reviewed and optionally modified for conflicting changes, it is ready for the final merge. The result of the final merge is a new source member with the combined changes, ready to add to the site source management tool such as SCLM.

Merge Files can be generated for individual list members, using Line Commands, or for all members in the list, using Group Actions. The file generation can run in the Foreground or as a Batch Job, depending on the value in the Exec Mode field. This section describes the Foreground execution process. For a description of Batch mode execution, see Chapter 6, “The Merge Tool batch utility,” on page 55.

Note: When using Group Actions, you can exclude selected members from the list. See “Using Group Actions” on page 25.

There are two variations of the command that generates a Merge File. These are:

GM Generate Merge File.

This command creates the Merge File from the Work File member. It is used when you have modified the Work File to resolve conflicts and ensures that the Merge File is created using your modifications.

GB Generate Work and Merge File.

This command creates both the Work and Merge Files in one action. As there is no opportunity to modify the Work File during the process, it is most commonly used when you have already examined a Work File and have determined that there are no conflicts or when you have resolved all conflicts by modifying the File1 or File2 members, not the Work File member.

Note: If you have previously run a GW command and then modified the Work File, running this command replaces the modified Work File.

When addressing conflicts in the Work File, you might determine that the change, although in the same section of code, does not overlap. You might also determine that all the changes are to be merged, even lines in conflict.

Generating the Merge File from a Work File

To generate a Merge File from an existing Work File (that might contain modifications needed to resolve conflicts):

1. Build your Work File, using the techniques described in “Generating the Work File” on page 37.
2. Modify your Work File, using the techniques described in Chapter 4, “Review changes and address conflicts,” on page 39
3. Set the Execution Mode field to 1. Foreground. For information about using Batch execution, see Chapter 6, “The Merge Tool batch utility,” on page 55.
4. Type the GM command in the Line Command field adjacent to the relevant list member.

OR

Type the GM command in the Group Action field. This generates a Merge File

Generating the Merge File

for all included members.

OR

Type a "/" into the Line Command or Group Action field. Select GM from the pop-up list of commands.

- Press Enter. The Merge File generation begins and a Progress Status message is displayed in a pop-up window.

```

IBM Merge Tool for z/OS and OS/390      Row 1 to 9 of 9
----- Progress Indicator -----      "/" to select option.
M                                     Source Type _ Preferences
-                                     -----
-                                     1. Foreground 2. Batch)
"                                     Source
G                                     Type
                                     ASM____
--- AZZCBLA1      YES    YES    YES    YES    COBOL__
--- AZZCBLA2      YES    YES    YES    YES    COBOL__
--- AZZCBL01      YES    YES    YES    YES    COBOL__
--- AZZCBL02      YES    ----- YES    YES    COBOL__
--- AZZCBL01      ----- YES    YES    YES    COBOL__
--- AZZCBL02      ----- ----- YES    YES    COBOL__
--- AZZCBL03      ----- ----- YES    YES    COBOL__
--- AZZCBL04      ----- ----- YES    YES    COBOL__
--- AZZCBL05      ----- ----- YES    YES    COBOL__
--- AZZCBL06      ----- ----- YES    YES    COBOL__
--- AZZCBL07      ----- ----- YES    YES    COBOL__
--- AZZCBL08      ----- ----- YES    YES    COBOL__
--- AZZCBL09      ----- ----- YES    YES    COBOL__
***** Bottom of data *****
Command ==>>> _____ Scroll ==>>> PAGE

```

Figure 59. Progress Status Message - Merge

- When the process is completed, each selected member contains "YES" in the Have Merge column and "PROCESSED" in the Action column.

Generating the Work and Merge Files

To generate the Work and Merge Files from the Base File1 and File2 members:

- Build your list of members, using the techniques described in Chapter 2, "Identify the components to be examined," on page 11.
- Type the GB command in the Line Command field adjacent to the relevant list member.

OR

Type the GB command in the Group Action field. This generates a Work File and Merge File for all included members.

OR

Type a "/" into the Line Command or Group Action field. Select GB from the pop-up list of commands.

- Press Enter. The Work File generation begins and a Progress Status message is displayed in a pop-up window. The Merge File generation follows immediately, as indicated in the Progress Status message.
- When the process is completed, each selected member contains "YES" in the Have Work and Have Merge columns and "PROCESSED" in the Action column.

Merge outputs

Upon successful completion of the Generate Merge action the resulting member is placed in the Merge File. If the member already exists, it is replaced with the new Merge File member. The member name is the same as the Work File member name.

The output is the combined merge file. It contains the base (original) source with both sets of changes incorporated into the base, based on the merge directives in columns one through 5 of the Work File.

The Merge File members can then be added to the site source management tool and compiled and linked, if appropriate.

Test outputs

Although Merge Tool cannot assist in the testing of the merge output it is important to note that if the user determines that the testing is unsuccessful, they can perform the Work File edit, or even creation, and subsequent steps over and over until testing is successful.

Chapter 6. The Merge Tool batch utility

You can execute the Merge Tool file generation processes via a batch job. Using a batch job allows you to create, edit, store and repeat varying instructions for generating your Work, Merge and Report files.

Merge Tool DD statements

The JCL statements in the batch file use a number of DD names, to which you must allocate your libraries.

Table 8. Utility DD Statements

DDNAME	Description
CIGBASE	The Base File or original source used for modifications. This DD is required for the GW and GB actions. You can allocate an existing data set or allocate NULLFILE. If NULLFILE is allocated, the File1 data set is used in the generation.
CIGDD01	File1 source library. The first of up to two files containing the modified source. In the case of an emergency fix, it might be the location of the fixed production source. This DD is required for the GW, SS and GB actions. You can allocate an existing data set or allocate NULLFILE. If NULLFILE is allocated, the Base data set is used in the generation. If Base is also set to NULLFILE, the File2 data set is used. If File2 is also set to NULLFILE, no output is produced.
CIGDD02	File2 source library. The second, and optional, PDS containing the modified source. In the case of an emergency fix, it might be the location of the new release or enhanced source. This DD is required for the GW, SS and GB actions. You can allocate an existing data set or allocate NULLFILE. If NULLFILE is allocated, the File1 data set is used in the generation.
CIGWORK	The Work File library. This is where Merge Tool puts the output from the Generate Work action or where the Merge Tool gets the input for the Generate Merge action. This DD is required for the GW, GM and GB actions. You must allocate an existing data set.
CIGMERGE	The Merge File library. This is where Merge Tool puts the output from the final merge step (Generate Merge). This DD is required for the GM and GB actions. You must allocate an existing data set.
CIGSTATS	Member Statistics report, showing the number of modifications for each source module processed. Conflicting modifications are of particular significance. To produce the report, the DD is required and can be allocated a data set or SYSOUT.
CIGSUMM	The Summary Statistics report, showing totals related to number of members processed. To produce the report, the DD is required and can be allocated a data set or SYSOUT.
CIGLOG	The Log Messages report, showing parameters and statements in effect during the merge process. If processing fails, the reason is reported here. To produce the report, the DD is required and can be allocated a data set or SYSOUT.
CIGIN	Specifies members to be acted upon.

Running in batch mode

When you choose to run the GW, SS, GM or GB commands in batch mode, Merge Tool generates the appropriate JCL. It uses the libraries and file types you specified in the session settings at the time of adding the member to create the default DD allocations and SELECT statements. If an Input library is not specified, the JCL allocates NULLFILE to the DD Name. If a Report Output library is not specified, the relevant line is omitted from the JCL.

Once generated, you can submit the default JCL or you can edit it and then submit your edited version. The JCL is created in a temporary data set, which is overwritten each time that you issue a new GW, SS, GM or GB command. If you want to keep your edited version for future use, you can copy the text to your own data set location.

Using batch mode with the default JCL

To generate the Work, Merge and Report files, using the default JCL, you must:

1. Build your list of members, using the techniques described in Chapter 2, "Identify the components to be examined," on page 11.

Note: If you want to generate the statistics reports, you must have added your members to the list with an allocated data set library for each desired report type. (See "Specify the data set libraries" on page 11 for details.)

2. Set the Execution Mode field to 2. Batch. For information about using Foreground execution, see "Generating the Work File" on page 37.
3. Type GW, SS, GM or GB into the Group Action field or into one or more of the Line Command fields for individual members.

Note: If you enter a command in the Group Action field, all non-excluded members are included in the same generated JCL. If you enter multiple commands in the Line Command fields, each command generates its own JCL, which runs consecutively when submitted.

4. Press Enter. The JCL is generated and the Merge Tool Edit and Submit JCL panel (Figure 60 on page 57) is displayed.


```

Merge Tool Edit and Submit JCL

Option ==>

PROGRAM: AZZASM01

Select one of the following options and press enter:

          1  Edit JCL
          2  Submit JCL

Job cards:
=====> //USERID JOB (YOURACCT),'NAME',
=====> //   CLASS=A,REGION=2M,
=====> //   MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID

Additional JCL:
=====> _____
=====> _____
=====> _____
=====> _____
=====> _____

F1=HELP   F2=SPLIT   F3=END     F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP    F10=LEFT    F11=RIGHT  F12=RETRIEVE

```

Figure 60. Merge Tool Edit and Submit JCL panel

Note: The member name or "GROUP ACTION" is displayed under the Option line to help you identify what is being processed.

5. If required, modify the Job cards and Additional JCL fields.
6. Type 2 on the Option line and press Enter. The default JCL is submitted and a message showing the submission details is displayed.

```

IKJ56250I JOB SUEBURNC(JOB15945) SUBMITTED
***

```

Figure 61. JCL submit message

7. Press Enter to return to the Edit and Submit panel.
8. Press PF3 (End) to return to the main panel. The Action column contains SUBMITTED for all members processed.

Note: If you used multiple Line Commands, then next JCL is automatically submitted and the confirmation message is displayed. Press Enter and then PF3 repeatedly, until all line commands have been processed. When you return to the main panel, the Action column contains SUBMITTED for all affected members.

Edit and submit JCL

You can modify the default JCL generated by Merge Tool to specify your own requirements, such as writing the reports to SYSOUT instead of to a data set.

To edit and submit the JCL:

1. Follow steps 1 on page 56 to 5 to display the Merge Tool Edit and Submit JCL panel (Figure 60).

Running in batch mode

2. Type 1 on the Option line and press Enter. The default JCL is displayed in an ISPF Edit session.

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
ISREDDE2  SUEBURN.SPFTEMP1.CNTL                      Columns 00001 00072
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>         your edit profile using the command RECOVERY ON.
000001 //SUEBURNC JOB (YOURACCT),'NAME',
000002 //      CLASS=A,REGION=2M,
000003 //      MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID
000004 //*-----*
000005 //* IBM MERGE TOOL FOR Z/OS AND OS/390                *
000006 //*-----*
000007 //STEP1 EXEC PGM=IKJEFT01
000008 //STEPLIB DD DSN=AZZ.V1R1.SAZZLOAD,DISP=SHR
000009 //SYSPROC DD DSN=AZZ.V1R1.SAZZCLIB,DISP=SHR
000010 //SYSTSIN DD *
000011 EXEC 'AZZ.V1R1.SAZZCLIB(AZZMERGE) '
000012 //SYSTSPRT DD SYSOUT=*
000013 //*
000014 //* TO BUILD A WORK FILE,
000015 //* ALLOCATE CIGBASE, CIGDD01, CIGDD02, AND CIGWORK DD STATEMENTS
Command ==>                                           Scroll ==> PAGE
F1=Help      F2=Split      F3=Exit      F5=Rfind     F6=Rchange   F7=Up
F8=Down     F9=Swap       F10=Left    F11=Right   F12=Cancel
```

Figure 62. JCL in Edit session

3. The generated JCL reflects the command used on the main panel. You can modify this, as follows:

GW - Generate Work File

When using this command, you must allocate the CIGBASE, CIGDD01, CIGDD02, and CIGWORK DD statements to existing data sets. You can also specify the CIGSTATS, CIGSUMM and CIGLOG DD statements, however, these are not required. If specified, these statements are, by default, allocated to the data sets given in Set Libraries. However, you can modify the CIGSTATS, CIGSUMM and CIGLOG statements to allocate them to SYSOUT.

SS - Show Statistics for Member

When using this command, you must allocate the CIGBASE, CIGDD01, CIGDD02, CIGWORK and CIGSTATS DD statements to existing data sets. You can also specify the CIGSUMM and CIGLOG DD statements, however, these are not required. If specified, these statements are, by default, allocated to the data sets given in Set Libraries. However, you can modify the CIGSTATS, CIGSUMM and CIGLOG statements to allocate them to SYSOUT.

GM - Generate Merge File from Work File

When using this command, you must allocate the CIGWORK and CIGMERGE DD statements to existing data sets. You can also allocate the CIGLOG statement, however, this is not required. If specified, CIGLOG can be allocated to a data set or to SYSOUT.

Note: To create a Merge File directly from a Work File, specify the ddnames CIGWORK and CIGMERGE *only*. CIGBASE, CIGDD01, and CIGDD02 must be omitted. If CIGBASE, CIGDD01, and

CIGDD02 are specified, the Work File specified in CIGWORK is *replaced* prior to the creation of the final merged file.

GB - Generate Work File and Merge File

When using this command, you must allocate the CIGBASE, CIGDD01, CIGDD02, CIGWORK, and CIGMERGE DD statements to existing data sets. You can also allocate the CIGSTATS, CIGSUMM and CIGLOG statements, however, these are not required. If specified, these statements can be allocated to a data set or to SYSOUT.

4. Press PF3 to end the Edit session and save your changes. The Edit and Submit JCL panel is displayed again, with the edited member active.
5. To submit the edited JCL, type 2 on the Option line and press Enter. The JCL confirmation message is displayed.
6. Press Enter to return to the Edit and Submit JCL panel, with the current member still active.
7. Press PF3 to return to the main panel.

If you used multiple Line Commands, the next member is displayed in the Edit and Submit JCL panel. You can edit and submit each member, until all line commands have been processed. When you return to the main panel, the Action column contains SUBMITTED for all affected members.

Note: You can also submit your edited JCL directly from the Edit panel, by entering the SUB command on the Primary Command line.

The JCL

Three sample JCL members can be found in the **hlq.SAZZJCL** library created during the installation procedure. **hlq** denotes the high level qualifier of the product's target library. The batch process works out what action it is going to perform, based on the files that are allocated in the JCL.

AZZMRKUP - build a mark-up (Work) file

The following JCL is a sample of the AZZMRKUP member. Note the exclusion of the CIGMERGE DD statement in this sample JCL.

Include the CIGMERGE DD statement only if you wish to have the final merged file created.

```

:
//STEP1 EXEC PGM=IKJEFT01
//STEPLIB DD DSN=AZZ.SAZZLOAD,DISP=SHR
//SYSTSIN DD *
EXEC 'AZZ.SAZZCLIB(AZZMERGE)'
//SYSTSPRT DD SYSOUT=*
//*
/* TO BUILD A WORK FILE,
/* ALLOCATE CIGBASE, CIGDD01, CIGDD02, AND CIGWORK DD STATEMENTS
/*
/* NOTE: CIGWORK MUST HAVE A LRECL 7 LARGER THAN CIGDD01
/*
//CIGBASE DD DISP=SHR,DSN=AZZ.SAZZSAMB
//CIGDD01 DD DISP=SHR,DSN=AZZ.SAZZSAM1
//CIGDD02 DD DISP=SHR,DSN=AZZ.SAZZSAM2
//CIGWORK DD DISP=SHR,DSN=AZZ.SAZZSAMW
/*
/* IF BASE OR FILE 2 IS OMITTED THEN FILE1 NAME IS USED
/* IF FILE 1 IS OMITTED THEN BASE NAME WILL BE USED
/* IF FILE 1 AND BASE IS OMITTED THEN FILE 2 WILL BE USED
/* IF NO SELECT STATEMENT SPECIFIED THEN '*' WILL BE USED.
/*
/* TYPE CAN BE JCL, DATA, ASM, COBOL, PL1 OR XX:XX
/* DEFAULT TYPE IS DATA (ALL COLUMNS)
/*
//CIGIN DD *
SELECT FILE1 XXXXXXXX FILE2 XXXXXXXX WORK XXXXXXXX TYPE XXXXX
/*
//CIGLOG DD SYSOUT=*
/*
/* PRESENCE OF CIGSUMM WILL CAUSE SUMMARY STATISTICS
/* TO BE WRITTEN.
/*
/* PRESENCE OF CIGSTATS WILL CAUSE MEMBER LEVEL STATISTICS
/* TO BE WRITTEN.
/*
//CIGSUMM DD SYSOUT=*
//CIGSTATS DD SYSOUT=*

```

Figure 63. Build Work File

AZZMERG0 - build the final Merge File from the Work File

The following JCL is a sample of the AZZMERG0 member. Note the exclusion of the CIGBASE, CIGDD01 AND CIGDD02 DD statements.

Include the aforementioned DD statements only if you wish to create a new work file.

```

:
//STEP1 EXEC PGM=IKJEFT01
//STEPLIB DD DSN=AZZ.SAZZLOAD,DISP=SHR
//SYSTSIN DD *
EXEC 'AZZ.SAZZCLIB(AZZMERGE)'
//SYSTSPRT DD SYSOUT=*
//*
//* TO BUILD A MERGE FILE FROM A WORK FILE,
//* ALLOCATE CIGMERGE.
//*
//* NOTE: CIGWORK MUST HAVE A LRECL 7 LARGER THAN CIGDD01
//*
//CIGWORK DD DISP=SHR,DSN=AZZ.SAZZSAMW
//CIGMERGE DD DISP=SHR,DSN=AZZ.SAZZSAMB
//*
//*
//* IF BASE OR FILE 2 IS OMITTED THEN FILE1 NAME IS USED
//* IF FILE 1 IS OMITTED THEN BASE NAME WILL BE USED
//* IF FILE 1 AND BASE IS OMITTED THEN FILE 2 WILL BE USED
//* IF NO SELECT STATEMENT SPECIFIED THEN '*' WILL BE USED.
//*
//* TYPE CAN BE JCL, DATA, ASM, COBOL, PL1 OR XX:XX
//* DEFAULT TYPE IS DATA (ALL COLUMNS)
//*
//CIGIN DD *
SELECT FILE1 XXXXXXXX FILE2 XXXXXXXX BASE XXXXXXXX TYPE XXXXX
/*
//CIGLOG DD SYSOUT=*
//*
//* PRESENCE OF CIGSUMM WILL CAUSE SUMMARY STATISTICS
//* TO BE WRITTEN.
//*
//* PRESENCE OF CIGSTATS WILL CAUSE MEMBER LEVEL STATISTICS
//* TO BE WRITTEN.
//*
//CIGSUMM DD SYSOUT=*
//CIGSTATS DD SYSOUT=*

```

Figure 64. Build Final Merge File

AZZMRGIT - build a mark-up (Work) file and final Merge File

The following JCL is a sample of the AZZMRGIT member. Note the inclusion of the CIGBASE, CIGDD01, CIGDD02, CIGWORK and CIGMERGE DD statements in this sample JCL.

This JCL creates a Work File and immediately interprets the Work File, creating a Merged File with both sets of changes incorporated, including conflicting changes.

```

:
//STEP1 EXEC PGM=IKJEFT01
//STEPLIB DD DSN=AZZ.SAZZLOAD,DISP=SHR
//SYSTSIN DD *
EXEC 'AZZ.SAZZCLIB(AZZMERGE)'
//SYSTSPRT DD SYSOUT=*
//*
//* TO BUILD A WORK FILE,
//* ALLOCATE CIGBASE, CIGDD01, CIGDD02, AND CIGWORK
//* AND COMMENT OUT CIGMERGE
//*
//* TO BUILD A MERGE FILE FROM A WORK FILE,
//* ALLOCATE CIGMERGE.
//* AND COMMENT OUT CIGBASE, CIGDD01 AND CIGDD02
//*
//* NOTE: CIGWORK MUST HAVE A LRECL 7 LARGER THAN CIGDD01
//*
//CIGBASE DD DISP=SHR,DSN=AZZ.SAZZSAMB
//CIGDD01 DD DISP=SHR,DSN=AZZ.SAZZSAM1
//CIGDD02 DD DISP=SHR,DSN=AZZ.SAZZSAM2
//CIGWORK DD DISP=SHR,DSN=AZZ.SAZZSAMW
//CIGMERGE DD DISP=SHR,DSN=AZZ.SAZZSAMB
//*
//* IF BASE OR FILE 2 IS OMITTED THEN FILE1 NAME IS USED
//* IF FILE 1 IS OMITTED THEN BASE NAME WILL BE USED
//* IF FILE 1 AND BASE IS OMITTED THEN FILE 2 WILL BE USED
//* IF NO SELECT STATEMENT SPECIFIED THEN '*' WILL BE USED.
//*
//* TYPE CAN BE JCL, DATA, ASM, COBOL, PL1 OR XX:XX
//* DEFAULT TYPE IS DATA (ALL COLUMNS)
//*
//CIGIN DD *
SELECT FILE1 XXXXXXXX FILE2 XXXXXXXX WORK XXXXXXXX TYPE XXXXX
/*
//CIGLOG DD SYSOUT=*
//*
//* PRESENCE OF CIGSUMM WILL CAUSE SUMMARY STATISTICS
//* TO BE WRITTEN.
//*
//* PRESENCE OF CIGSTATS WILL CAUSE MEMBER LEVEL STATISTICS
//* TO BE WRITTEN.
//*
//CIGSUMM DD SYSOUT=*
//CIGSTATS DD SYSOUT=*

```

Figure 65. Create Merged File

The Merge Tool syntax

The Merge Tool syntax allows the user to specify the members to be acted upon. The following tables document the parameters and syntax for the build Work File and build Merge File actions.

For AZZMRGIT and AZZMRKUP the SELECT statement is as follows:

```
SELECT FILE1 XXXXXXXX FILE2 XXXXXXXX WORK XXXXXXXX TYPE XXXXX
```

For AZZMERG0 the SELECT statement is as follows:

```
SELECT FILE1 XXXXXXXX FILE2 XXXXXXXX BASE XXXXXXXX TYPE XXXXX
```

Figure 66. Syntax of Member Select Statements

The SELECT parameter identifies the objects of the merge process. The sub-parameters and their values are documented in the following table. There must be at least one SELECT statement contained within the CIGIN DD statement.

The syntax for Merge Tool requests is defined in the following table.

Table 9. Syntax for Merge Tool Requests

Parameter	Value/Default
FILE1	The name or name pattern of the members in File1 to be object of the action. This parameter is required.
FILE2	The name or name pattern of the members in File2 to be object of the action. This parameter is optional. Default is same member name as File1.
BASE	The name or name pattern of the members in the Base File to be the object of the action. This parameter is optional. Default is same member name as File1.
TYPE	JCL, DATA, ASM, COBOL, PL1, SS:EE indicates the type of source being merged. This overrides the default merge compare columns (all source columns) to ones specific to the type. Example: COBOL directs Merge Tool to compare source columns 7 through to 72. This parameter is optional. Default is all columns. Note: If a specific type is not specified then start and end columns can be specified. For example 1:72
WORK	Intermediate merged file member

Syntax examples

Example 1:

```
SELECT FILE1 *
```

Example1 requests all members that match the member names of FILE1 as objects to the action.

Example 2:

```
SELECT FILE1 AZZCBY* FILE2 AZZCBZ* BASE AZZCBL* TYPE COBOL
```

Example2 requests to use the member name pattern to match the member names of FILE1 that begin with AZZCBY and members in FILE2 that begin with AZZCBZ with BASE members that begin with AZZCBL. Additionally, the COLUMN TYPE was specified as COBOL so only columns 7 to 72 are compared.

Example 3:

```
SELECT FILE1 AZZCBL01
```

Example 3 requests that Merge Tool use the member name specified in the FILE1 parameter for the FILE2 and BASE files as object for the action.

The JCL

Appendix A. Quick reference for commands

Member actions

The Merge Tool uses a set of actions that can be executed against any member within the member list. The action is executed for that member only. These actions can be entered in the Line Command field or can be selected from a pop-up list. To select from the pop-up list, enter a "/" in the Line Command field.

The following table contains a list of Member Actions and their descriptions.

Table 10. Member Actions

Action	Description
GW	Generate Work File
SS	Show Statistics for Member
GB	Generate Both Work and Merge File
GM	Generate Merge File from Work File
E (or EB, E1, E2, EW, EM)	Edit Member (see note) <ul style="list-style-type: none">• EB - edit base file member• E1 - edit file1 member• E2 - edit file2 member• EW - edit work file member• EM - edit merge file member
B (or BB, B1, B2, BW, BM)	Browse Member (see note) The member is displayed in Browse or View mode, depending upon your Preference setting. <ul style="list-style-type: none">• BB - browse base file member• B1 - browse file1 member• B2 - browse file2 member• BW - browse work file member• BM - browse merge file member
X	Exclude Member
I	Include Excluded Member

Note: When the action "E" or "B" is entered, a panel is displayed prompting the user to select the library from which to perform that action.

Group actions

Some member actions can be performed against the entire member list and affect all non-excluded members. These actions can be entered in the Group Action field or can be selected from a pop-up list. To select from the pop-up list, enter a "/" in the Group Action field.

The following table contains a list of Group Actions and their descriptions.

Table 11. Group Actions

Action	Description
GW	Generate Work File
SS	Show Statistics for Member
GB	Generate Both Work and Merge File
GM	Generate Merge File from Work File

Primary commands

The Primary Commands act upon the member list itself and are entered in the Primary Command line.

Table 12. Primary Commands

Command	Function
CLEAR	Clears the member list area. Once cleared, the member list cannot be restored.
LOCATE or L	Locates a member in the list. Must be followed by a member name/pattern.
RESET or RES	Clears the "Action" column and restores any Excluded members.
X ALL	Places all members in Excluded mode.

Appendix B. Sample Merge Tool scenarios

The following usage scenarios are used to illustrate, from a programmer's perspective, the possible situations in which Merge Tool can facilitate application maintenance. An example application called "Online Ordering" is used in the usage scenarios. In all scenarios, the customer must install and configure Merge Tool.

Note: For this and other scenarios that follow, all ISPF actions are also available in a batch utility. It is up to the user to determine if foreground execution of Merge Tool is a viable option for the project, based on resources and the scope of the merge effort. When attempting to merge the contents of entire libraries, use the batch utility.

Scenario #1 "Basic Parallel Development" The Emergency Production Fix

Programmer codes enhancement to the "Online Ordering" system main program. While the enhancement is being coded a bug is found and fixed in the current production source by the maintenance team.

Problem

The source for the enhancement does not contain the production fix. Promotion of the enhancement reverses the production fix, causing production failures to return.

Solution

Reconcile the fix into the enhanced version of the source. Prior to promotion the programmer must incorporate the fix into their code using Merge Tool.

Assumption

Merge Tool has been installed at the customer site into a set of ISPF libraries. User has created the statistic and log data sets required for reporting.

1. Identify the location of the:

Original or base source

This would be the common version of the source used for the enhancement as well as the production fix.

Fixed production source (File1)

This would be the version of the source currently running in production with the bug fix applied.

Enhanced source with new functionality (File2)

This would be the version of the source that contains the required enhancement to the application.

2. Invoke Merge Tool and configure the tool for the session:
 - a. Review the session preferences for the marked-up work file.
 - b. Update the Library data sets for the session based on 1.
3. Specify member name and build member list by pressing Enter.
4. Select member with "/" for action list and "1" to generate the Work file
 - This can also be accomplished by typing GW across from the member on the main Merge Tool panel.

Scenario #1 "Basic Parallel Development"

- If the changes are of a simple nature, the programmer can use the GB action to generate the Work file and the Merge file, in one step, skipping the next three steps.
 - 5. At this point the user can review the member statistics or immediately view the work file for conflicting changes.
 - 6. The programmer now edits the Work file to address any conflicting changes made to both versions of the code. The edit session is invoked with the EW action or "/" for the action list and 5 followed by 4 on the next two panels.
 - a. Source lines can be inserted into or deleted from the work file.
 - b. Merge directives can be changed (within columns 1 through 5 of the work file).
 - c. All conflicting lines can remain and be included in Merge output.
 - 7. Once the programmer is finished reviewing and addressing any conflicting changes, the Work file is used to build a consolidated member in the Merge file. The merge is invoked with the GM action or "/" for the action list and "3" for Generate Merge file from Work file.
 - 8. Once the source is consolidated, the programmer can access their site source management tool (such as SCLM), and introduce the consolidated member, building new output to test.
 - a. Source is added from the Merge file and outputs are built.
 - b. Outputs are tested for both the production fix and the enhancement.
- Note:** If any of the tests fail the Work file can be edited and merged again, creating new output to test (steps 6, 7 and 8). These steps are repeated until successful testing is accomplished.
9. Once the programmer has successfully tested the source for both the fix and enhancement the code can be safely promoted through the life cycle and ultimately to production.

Scenario #2 "Common Parallel Development" Two programmers, one source

Two programmers are assigned to work on different bugs in the same program. Rather than one programmer waiting until the other programmer has completed their changes, and successfully tested the program, they begin development at the same time.

Problem

Each version of the fixed program is missing the source for the 'other' fix. Independent promotion of the fixed code causes source regression and production failures to return.

Solution

Programmers must incorporate the independently coded fixes, using Merge Tool. Once incorporated, they must test for both conditions and promote to production.

Assumption

Merge Tool has been installed at the customer site into a set of ISPF libraries. User has created the statistic and log data sets required for reporting.

1. Programmers must determine what source is identified as File1 and what source is identified as File2. Identify the location of the:

Scenario #2 "Common Parallel Development"

Original or base source

This would be the common version of the program both programmers used for their source.

Source with first fix (File1)

This would be the version of the source with one of the fixes for the source currently running in production.

Source with second fix (File2)

This would be the version of the source with the other fix for the source currently running in production.

2. Invoke Merge Tool and configure the tool for the session:
 - a. Review the session preferences for the marked-up work file.
 - b. Update the Library data sets for the session based on 1.
 3. Specify member name and build member list by pressing Enter.
 4. Select member with "/" for action list and "1" to generate the Work file
 - This can also be accomplished by typing GW across from the member on the main Merge Tool panel.
 - If the changes are of a simple nature, the programmers can use the GB action to generate the Work file and the Merge file, in one step, skipping the next three steps.
 5. At this point the programmers would review the member statistics to determine the complexity of the consolidation of development efforts, or immediately view the work file for conflicting changes (see the next step).
 6. The programmer now edits the Work file to address any conflicting changes made to both versions of the code. The edit session is invoked with the EW action or "/" for the action list and 5 followed by 4 on the next two panels.
 - a. Source lines can be inserted into or deleted from the work file.
 - b. Merge directives can be changed (within columns 1 through 5 of the work file).
 - c. All conflicting lines can remain and be included in Merge output.
 7. Once the programmer is finished reviewing and addressing any conflicting changes, the Work file is used to build a consolidated member in the Merge file. The merge is invoked with the GM action or "/" for the action list and "3" for Generate Merge file from Work file.
 8. Once the source is consolidated, the programmers can access the site source management tool (such as SCLM) and introduce the consolidated member, building new output to test.
 - a. Source is added from the Merge file and outputs are built.
 - b. Outputs are tested for both the production fix and the enhancement.
- Note:** If any of the tests fail the Work file can be edited and merged again, creating new output to test (steps 6, 7 and 8). These steps are repeated until successful testing is accomplished.
9. Once the programmers have successfully tested the source for both conditions the code can be safely promoted through the life cycle and ultimately to production.

Scenario #3 "Complex Parallel Development" Release management in-house

A Company must maintain an existing software release while developing a new release. The maintenance team must continue to support and code against an existing release, while the development team works on a new release of the product.

Problem

Because current production source was the base of the new release, eventually the maintenance changes must be incorporated in the next release. Promotion of the new release without incorporating the maintenance reverses the production maintenance, causing production failures to return.

This scenario is more complex than the others are because many of the details are unknown. Proper project management requires these details such as number of members, complexity of the changes, and conflicting changes when forecasting the resources needed for the reconciliation effort.

Solution

Using Merge Tool, Project Management can assess the development effort required to complete the consolidation of maintenance applied since coding of the new release began. The development team can then reconcile the maintenance into the new version of the source using Merge Tool.

Assumption

Merge Tool, has been installed at the customer site into a set of ISPF libraries. User has created the statistic and log data sets required for reporting.

Project management actions follow:

1. Identify the location of the:

Original or base source

This would be the common version of the source used for the enhancement as well as the production fix.

Maintained production source (File1)

This would be the version of the source currently running in production with the bug fix applied.

New release of product (File2)

This would be the version of the source that contains the required changes to the application, creating the next release.

2. Invoke Merge Tool and configure the tool for the session
 - a. Review the session preferences for the marked-up work file.
 - b. Update the Library data sets for the session based on 1.
3. Enter "*" in member name, build member list by pressing Enter.
4. Run member statistics, and summary statistics against the appropriate files.

Note: When using Foreground Execution and your members are spread across several data set types (for example, COBOL and Copybooks in one set of libraries, JCL and PROC source in another set of libraries), you need to update the Library data sets between each submit. In Batch JCL this is a change to the CIGDD01, CIGDD02 and CIGWORK DD statements of the Merge Tool JCL.

Scenario #3 "Complex Parallel Development"

5. Analyze the output of the reports to determine the following information:
 - a. Number of members requiring consolidation
 - b. Complexity of change conflict
 - c. Complexity of changes to each version of source (that is, inserts, deletes, and conflicting changes).
6. Based on analysis Project Management assigns merge tasks to programmers.

Programmer actions follow:

1. Obtain the location of the:
 - Original or base source**

This would be the common version of the source used for the enhancement as well as the production fix.
 - Maintained production source (File1)**

This would be the version of the source currently running in production with the bug fix applied.
 - New release of application (File2)**

This would be the version of the source that contains the required changes to the application, creating the next release.
2. In TSO invoke Merge Tool and configure the tool for the session:
 - a. Review the session preferences for the marked-up work file.
 - b. Update the Library data sets for the session based on 6.
3. Enter "*" in member name, build member list by pressing Enter.
4. Select GROUP ACTION with "/" for action list and "1" to generate the Work file
 - This can also be accomplished by typing GW in the Group Action field on the main Merge Tool panel.
 - If the changes are of a simple nature, the programmers can use the GB group action to generate the Work file and the Merge file, in one step, skipping the next three steps.
5. Programmer reviews member statistics and determines the complexity of the effort to consolidate the two versions and, which members require additional resources such as contact with the maintenance programmer for details. The programmer could choose to immediately view the work file for conflicting changes.
6. The programmer now edits the Work file to address any conflicting changes made to both versions of the code. The edit session is invoked with the EW action or "/" for the action list and 5 followed by 4 on the next two panels.
 - a. Source lines can be inserted into or deleted from the work file.
 - b. Merge directives can be changed (within columns 1 through 5 of the work file).
 - c. All conflicting lines can remain and be included in Merge output; this can be done to test to see if both sets of changes can coexist.
7. Once the programmer is finished reviewing and addressing any conflicting changes, the Work file is used to build a consolidated member in the Merge file. The merge is invoked with the GM action or "/" for the action list and "3" for Generate Merge file from Work file.
8. Once the source is consolidated, the programmers can access the site source management tool (such as SCLM) and introduce the consolidated member, building new output to test.

Scenario #3 "Complex Parallel Development"

- a. Source is added from the Merge file and outputs are built.
- b. Outputs are tested for both the production fix and the enhancement.

Note: If any of the tests fail the Work file can be edited and merged again, creating new output to test (steps 6 on page 71, 7 on page 71 and 8 on page 71). These steps are repeated until successful testing is accomplished.

9. Once the programmers have successfully tested the source for all conditions the code can be safely promoted with the rest of the new release through the life cycle and ultimately to production.

Scenario #4 "Complex Parallel Development" Vendor customized source

A Company purchased an application package (Version 1) from a software vendor. Consultants were hired to make modifications to the software. When the vendor releases a version upgrade (Version 2) the company must assess the resources required for retrofitting the customization to the upgrade. Once analysis is complete the company must incorporate their customization into the upgraded vendor source.

Problem

The consultants are no longer on this project. With them went the detailed knowledge of the customization done to this application. The company must determine if there is a conflict between the customization and the changes of the new version and then the customization must be re-coded into the new release of the vendor software.

This scenario is complex. Many of the details are unknown. Accurate project management requires these details such as number of members, complexity of the changes, and conflicting changes when forecasting the resources needed for the incorporation effort. Additionally, you want to avoid having to entirely re-code the customization.

Resolution

Using Merge Tool, Project Management can accurately assess the development effort required to address conflicting code changes and complete the retrofitting of the customization with the new release. The development team can then easily reconcile the customization to the new version, using Merge Tool.

Assumption

Merge Tool has been installed at the customer site into a set of ISPF libraries. User has created the statistic and log data sets required for reporting.

1. Unload all Vendor source to site libraries.
2. Determine which types of source libraries contain customized vendor source.

Note: The following steps can be done for each of the sets of libraries where customization has taken place. If the user prefers not to perform these steps multiple times, the ISPF part of Merge Tool allows the user to append to lists of members and override member type (for example, COBOL, Assembler etc.) indicating compare columns. Keep in mind the reports could become quite large and unmanageable, making reporting library by library a more digestible quantity.

Project Management actions follow:

1. Identify the location of the:

Original or base source

This would be the previous version (Version 1) of the vendor source without the customization.

Production source (File1)

This would be the customized Version 1 source, currently running in production.

New release of product (File2)

This would be the source for Version 2 (the upgrade) the vendor shipped (requiring the retrofit of the customization).

2. Invoke Merge Tool in TSO and configure the tool for the session:
 - a. Review the session preferences for the marked-up work file.
 - b. Update the Library data sets for the session based on 1.
3. Setting member value to "*", build member list by pressing Enter.
4. Run member statistics, and summary statistics against the appropriate files.

Note: When using Foreground Execution and your members are spread across several data set types (for example, COBOL and Copybooks in one set of libraries, JCL and PROC source in another set of libraries), you need to update the Library data sets between each submit. libraries, JCL and PROC source in another set of libraries, In Batch JCL this is a change to the CIGDD01, CIGDD02 and CIGWORK DD statements of the Merge Tool JCL.

5. Analyze the output of the reports to determine the following information
 - a. Number of members requiring consolidation
 - b. Complexity of change conflict
 - c. Complexity of changes to each version of source (that is, inserts, deletes, and conflicting changes).
6. Based on analysis Project Management assigns merge tasks to programmers.

Programmer actions follow:

1. Review the location of the following libraries:

Original (base) source

This would be the previous version (Version 1) of the vendor source without the customization.

Production source (File1):

This would be the customized Version 1 source, currently running in production.

New release of product (File2)

This would be the source for Version 2 (the upgrade) the vendor shipped requiring the retrofit of the customization.

2. Invoke Merge Tool in TSO and configure the tool for the session
 - a. Review the session preferences for the marked-up work file.
 - b. Update the Library data sets for the session based on 6.
3. Setting member value to "*", build member list by pressing Enter.
4. Select GROUP ACTION with "/" for action list and "1" to generate the Work file.

Scenario #4 "Complex Parallel Development"

- This can also be accomplished by typing GW in the Group Action field on the main Merge Tool panel.
 - If the changes are of a simple nature, the programmers can use the GB group action to generate the Work file and the Merge file, in one step, skipping the next three steps.
5. Programmer reviews Work File and statistics and determines which members require retrofitting and the complexity of the effort to consolidate the two versions.
 6. The programmer then edits the Work file members to address any conflicting changes made to both versions of the code. The edit session is invoked with the EW action or "/" for the action list and 5 followed by 4 on the next two panels.
 - a. Source lines can be inserted into or deleted from the work file.
 - b. Merge directives can be changed (within columns 1 through 5 of the work file).
 - c. All conflicting lines can remain and be included in Merge output; this might be done to test to see if both sets of changes can coexist.

Note: Changes made by File2 are the changes for the new release and in most cases remain. These changes can be ignored unless they overlap the customization (File1).

7. Once the programmer is finished reviewing and addressing any conflicting changes, the edited Work file members are used to build members into the Merge file. The merge action is invoked with the GM action or "/" for the action list and "3" for Generate Merge File from Work File.
8. Once the source is merged, the programmers can access the site source management tool (such as SCLM) and introduce the consolidated members, building new outputs to test.
 - a. Source is added from the Merge file and outputs are built.
 - b. Outputs are tested for both the customization and new release functionality.

Note: If any of the testing fails, the Work file can be edited and merged again, creating the new outputs to test (steps 6, 7 and 8). These steps are repeated until successful testing is accomplished.

9. When the programmers have successfully tested the source for all functionality, the code can be safely promoted with the rest of the new release through the life cycle and ultimately to production.

Appendix C. The Merge Tool customization process

Follow these steps in the order that they are presented. Once you have successfully completed all of the steps and executed all IVP tasks, you are ready to use the product. The value **hlq.** is used to denote the high level qualifier for the product's installed target libraries. In the samples supplied by the product, an hlq of AZZ is used within the sample application.

1. Modify member AZZISPF in the **hlq.SAZZCLIB** (Clist library)

Change the following data set names based on your SMP/E installation:

```
hlq.SAZZPENU (ISPF panel library)
hlq.SAZZMENU (ISPF message library)
hlq.SAZZSENU (ISPF skeleton library)
hlq.SAZZCLIB (ISPF clist and REXX source library)
```

2. Submit sample member AZZBLD1 in the **hlq.SAZZJCL**.

This creates the sequential data sets and partitioned data sets that are used with the sample application supplied with this product. It uses the following members from **hlq.SAZZJCL** to create the sequential data sets.

- AZZSAM1
- AZZSAM2
- AZZSAMB
- AZZSAMB
- AZZSAMB
- AZZSAMB
- AZZSAMB
- AZZSAMB
- AZZSAMB
- AZZSAMB
- AZZSAMB
- AZZSAMB

Consult the instructions in the sample job for more information.

3. Modify member AZZFRS02 in the **hlq.SAZZSENU** (ISPF skeleton library)

Change the following data set names based on your SMP/E installation:

```
hlq.SAZZCLIB (ISPF clist and REXX source library)
hlq.SAZZLOAD (load library)
```

4. Add the SAZZLOAD library to your TSO logon proc or as a link listed library.

Note: If you also have Breeze or Cloud 9 installed, please ensure that the Merge Tool load library is concatenated after the Breeze and Cloud 9 load libraries. There is a CIGINI module in the Merge Tool product that will cause a problem with Breeze and Cloud 9 if it is picked up first in the concatenation.

- a. TSO Logon Procedures

TSO logon procs are normally found in SYS1.TSOPROC.

If you are going to provide access to Merge Tool directly from an ISPF panel then you need to add the **hlq.SAZZCLIB** (CLIST library) to the TSO logon SYSPROC DD name concatenation.

If you do not link list the **hlq.SAZZLOAD** (load library) then you must add **hlq.SAZZLOAD** to the TSO logon proc STEPLIB.

To activate a newly created logon proc use the RDEFINE TSOPROC AZZV1R0 UACC(READ) command where AZZV1R0 is the name of the logon proc. In addition, the TSOPROC list must be refreshed using the following command: SETROPTS RACLIST(TSOPROC) REFRESH.

The Merge Tool customization process

b. Link listed library

Link listed libraries are normally found in SYS1.PARMLIB as member PROG00. Add the library **hlq.SAZZLOAD** as a LNKLIST statement. Issue the following commands to refresh the link list libraries:

```
SET PROG=00
```

```
F LLA,REFRESH
```

Note: PROG=00 would match the last two characters of the member name containing the LNKLIST definition.

If you are going to provide access to Merge Tool directly from an ISPF panel then you need to add the **hlq.SAZZCLIB** (CLIST library) to the TSO logon SYSPROC DD name concatenation.

5. Update the main ISPF panel to include easy access to Merge Tool

The primary ISPF panel is normally called ISR@PRIM, and is normally located in SYS1.LOCAL.ISPFPNLS or SISPPENU. This latter data set has a high level qualifier of ISP.

The following illustrates the ISPF panel changes to add the IBM Merge Tool as selection item 12 on the primary ISPF panel:

```
)AREA SAREA39
0 Settings      Terminal and user parameters
1 View         Display source data or listings
2 Edit         Create or change source data
3 Utilities    Perform utility functions
4 Foreground   Interactive language processing
5 Batch        Submit job for language processing
6 Command      Enter TSO or Workstation commands
7 Dialog Test  Perform dialog testing
8 LM Facility  Library administrator functions
9 IBM Products IBM program development products
10 SCLM        SW Configuration Library Manager
11 Workplace   ISPF Object/Action Workplace
12 Merge Tool  IBM Merge Tool for z/OS and OS/390
M More         Additional IBM Products
)INIT

IF (&ZSAR = 'CAL','UPS','SES') &ZCMD = 'SP'
&ZSEL = TRANS (TRUNC (&ZCMD, '.'))
0, 'PGM(ISPISM) SCRNAME(SETTINGS)'
1, 'PGM(ISRBRO) PARM(ISRBRO01) SCRNAME(VIEW)'
2, 'PGM(ISREDIT) PARM(P,ISREDM01) SCRNAME(EDIT)'
3, 'PANEL(ISRUTIL) SCRNAME(UTIL)'
4, 'PANEL(ISRFPA) SCRNAME(FOREGRND)'
5, 'PGM(ISRJB1) PARM(ISRJPA) SCRNAME(BATCH) NOCHECK'
6, 'PGM(ISRPTC) SCRNAME(CMD)'
7, 'PGM(ISPYXDR) PARM(&ZTAPPLID) SCRNAME(DTEST) NOCHECK
8, 'PANEL(ISRLPRIM) SCRNAME(LMF)'
9, 'PANEL(ISRDIIS) ADDPOP'
10, 'PGM(ISRSCLM) SCRNAME(SCLM) NOCHECK'
11, 'PGM(ISRUDA) PARM(ISRWORK) SCRNAME(WORK)'
12, 'CMD(AZZISPF)'
M, 'PANEL(IBMPRODS)'
X, EXIT
SP, 'PGM(ISPSAM) PARM(PNS)'
' ' ' '
' ' ' '
*,'?' )
&ZTRAIL=.TRAIL
```

Figure 67. Primary ISPF Panel

Appendix D. Merge Tool messages

Messages with a suffix of **I** are informational where a message with a **E** suffix is an error.

AZZF011E No member(s) found

Explanation: User specified a member name or member mask. No match was found for member or mask supplied.

User Response: Check libraries and member name (or mask) supplied.

AZZF012E Enter member mask

Explanation: Group action was specified with no members listed. No members selected for processing.

User Response: Build member list by specifying member name or mask and pressing enter. Then the user can specify a group action.

AZZF013E Invalid command

Explanation: User specified an invalid Merge Tool TSO command at the command line.

User Response: User must specify Clear, Reset, Exclude or Locate. These are the only Merge Tool commands recognized.

AZZF014E Stats DSN Needed

Explanation: User has specified SS to build a Work File member, however the member statistics file is not defined to the session.

User Response: Select "Set Libraries" and specify a data set for Member statistics.

AZZF015E Empty File

Explanation: User specified an empty file for File1, Merge Tool is unable to look at empty files.

User Response: Either put a dummy member in the offending data set or select "Set Libraries" and specify a data set with at least one member.

AZZF016E Locate what?

Explanation: User specified the LOCATE command but did not specify a full or partial member name to locate.

User Response: Specify the full member name or leading characters of member to locate.

AZZF017E &TBFUNC failure

Explanation: There was a failure during ISPF Table handling. RC= &TBRC, error occurred for &TBFUNC of &TBNAME.

User Response: Contact local administrator support.

AZZF018E Mutually exclusive

Explanation: User specified both source type and start/end columns.

User Response: User must supply either a valid Sort Type or start and end compare columns. The valid values are: COBOL, ASM, DATA, JCL, PL1 or start column, colon, end column.

AZZF019E Out of Range

Explanation: User specified an end column that is not greater than the start column. The end column must be greater than start column.

User Response: User must specify an end column that is greater than the start column. Only the bytes between (and including) start and end positions are compared.

AZZF020E Invalid file1 dataset

Explanation: User specified a sequential data set (DSORG=PS) as File1. File1 must be a PDS (DSORG=PO).

User Response: Specify correct data set in set libraries panel and resubmit the request.

FST0003I error-level = message-text

Explanation: Informational message: This message is written out when tracing is requested. The return code will be shown followed by a descriptive message following the equal sign.

User Response: none

FST0004I error-level = message-text

Explanation: Informational message: This message is written out when tracing is requested. The return code will be shown followed by a descriptive message following the equal sign.

User Response: none

FST0022E READ ERROR OCCURRED DURING PARSING OF FILE file-name

Explanation: An attempt to parse the syntax contained in the file shown in the message failed.

User Response: Ensure the file is defined as variable or fixed block (RECFM= parameter). Ensure the file is properly allocated to your session or to the JCL of the job being executed. Refer to the Merge Tool User's Guide for an explanation of the program being executed.

FST0023E DURING PARSING, UNABLE TO ACQUIRE REQUIRED STORAGE

Explanation: Insufficient storage was available to complete the parsing operation.

User Response: First attempt to increase the REGION= parameter on your job card or on the step being executed. For requests involving very large amounts of data, it might be necessary to reduce the amount of data processed via selection criteria. Refer to The Merge Tool ISPF Workbench for an explanation of selection criteria associated with the program being executed.

FST0024E EXPECTING KEYWORD: syntax-keyword

Explanation: A syntax error was encountered when parsing the input syntax file.

User Response: The syntax is described in The Merge Tool syntax Correct the syntax in error and rerun the job.

FST0025E EXPECTING VARIABLE TO FOLLOW KEYWORD: syntax-variable

Explanation: A syntax error was encountered when parsing the input syntax file.

User Response: Review the syntax is described in The Merge Tool syntax. Correct the syntax in error and rerun the job.

FST0026I label-1 @ label-2 DATA=text

Explanation: Informational message: This is a tracing message which appears when parser tracing is in effect.

User Response: none

FST0100E NO STORAGE AVAILABLE TO ALLOCATE FILE HANDLE xxxxxxxx

Explanation: The application could not allocate an internal file block because of storage problems.

User Response: Increase the REGION= parameter on your jobcard or step and rerun job.

FST0101E FAILURE TO SET UP FILE HANDLE FOR file-name

Explanation: The application could not complete the creation of an internal file block. This error could occur if the application could not allocate the data set associated with the ddname shown in the message.

User Response: Ensure the data set associated with the file shown in the message exists. If the file is a system generated ddname then ensure that the application database has been allocated and initialized. You might also need to increase your REGION= parameter on the jobcard or on the step being executed.

FST0110E INVALID FILE HANDLE PASSED TO \$\$QSAM

Explanation: This message indicates an internal error occurred when the application was attempting to process a QSAM file.

User Response: Check log for other related messages. If problem persists, call IBM Technical Support.

FST0111E CLOSE FAILURE FOR FILE file-name

Explanation: This message is issued from \$\$QSAM services. \$\$QSAM CLOSE request has failed.

User Response: Check log for other related messages. If problem persists, call IBM Technical Support.

FST0112E OPENO REQUESTED, BUT file-name IS ALREADY OPENED FOR OUTPUT

Explanation: This message is issued from \$\$QSAM services. Attempting to open data set that is already opened for output.

User Response: Check log for other related messages. Make sure you do not have the same file allocated as an output data set for two different ddnames.

FST0113E OPENI REQUESTED, BUT file-name IS ALREADY OPENED FOR INPUT

Explanation: This message is issued from \$\$QSAM services. Attempting to open data set that is already opened for input.

User Response: Check log for other related messages.

FST0114E ATTEMPTING TO OPEN file-name , BUT UNABLE TO ACQUIRE A FILE HANDLE

Explanation: This message is issued when attempting to acquire storage.

User Response: Check log for other related messages. You might also need to increase your REGION= parameter on the jobcard or on the step being executed.

FST0115E UNABLE TO ACQUIRE STORAGE TO HOLD THE DCB FOR file-name

Explanation: This message is issued when attempting to acquire storage, but insufficient storage existed to perform the operation.

User Response: Check log for other related messages. You might also need to increase your REGION= parameter on the jobcard or on the step being executed.

FST0116E FILE file-name IS NEITHER SEQUENTIAL OR PARTITIONED

Explanation: The file name shown should be sequential or partitioned, but was found to be some other type of DSORG.

User Response: Probable user error. Check your file allocations. This message means that a data set with other than PS or PO DSORG has been allocated to a PS or PO type ddname. Ensure all files are properly allocated. Fix problem and rerun the job.

FST0117E ATTEMPTING TO OPEN PDS AS A SEQUENTIAL FILE FOR file-name

Explanation: Merge Tool has encountered an output data set that is partitioned but does not have member name.

User Response: Check your data set allocation to ensure a member name is specified. Ensure all files are properly allocated. Fix allocations and rerun the job.

FST0118E UNABLE TO ACQUIRE BUFFER OF buff-len BYTES FOR FILE file-name

Explanation: This message is issued when attempting to acquire storage, but insufficient storage existed to perform the operation.

User Response: Check log for other related messages. You might also need to increase your REGION= parameter on the jobcard or on the step being executed.

FST0119E OPEN ISSUED FOR FILE file-name , BUT REQUEST FAILED

Explanation: Unable to open a QSAM file.

User Response: Check log for other related messages. Verify JCL is properly specified.

FST0120E AUTO-OPEN FOR file-name FAILED ON READ REQUEST

Explanation: Unable to open a QSAM file.

User Response: Check log for other related messages. Verify JCL is properly specified.

FST0121E AN I/O ERROR OCCURRED DURING A READ ON FILE file-name

Explanation: A read or write QSAM failure occurred.

User Response: Check log for other related messages. Verify JCL is properly specified. Check JES log for related errors on the DASD volume.

FST0122E BUFFER SIZE (buf-len) IS TOO SMALL TO HOLD RECORD (rec)

Explanation: Attempting to write a record to QSAM file with too small record size. The region might be too small to acquire a buffer needed to complete the request.

User Response: Check log for other related messages. Check data set LRECL. Make sure data sets have the LRECL required as per the manual. Increase the REGION= parameter on the jobcard or step being executed.

FST0124E AUTO-OPEN FOR file-name FAILED ON WRITE REQUEST

Explanation: Unable to open file shown in message.

User Response: Check log for other related messages. Make sure all required data sets are allocated as per the manual.

FST0125E WRITE TO file-name ISSUED, BUT FILE OPENED FOR INPUT

Explanation: Attempting to write to a QSAM file that is already opened for input.

User Response: Check log for other related messages. Make sure you do not have the same data set allocated as both an input and an output data set.

FST0126E AN I/O ERROR OCCURRED DURING A WRITE ON FILE file-name

Explanation: An I/O error occurred during a QSAM write operation.

User Response: Check log for other related messages. Verify JCL is properly specified. Check JES log for related errors on the DASD volume.

FST0127E NO STORAGE AVAILABLE TO PERFORM WRITE FUNCTION ON FILE file-name

Explanation: Region size is probably too small.

User Response: Increase the REGION= parameter on the jobcard or step. If problem persists, it might be necessary to reduce the amount of data being processed via selection criteria as described in the Merge Tool User's Guide .

FST0128E READ ISSUED AGAINST file-name , BUT THE FILE ISN'T OWNED BY \$\$QSAM

Explanation: Attempting to perform a QSAM read against a non-QSAM file.

FST0129E READ ISSUED AGAINST file-name , BUT THE FILE IS \$CHAINED

Explanation: This message is issued from \$\$QSAM services.

User Response: This is an internal error. Check log for other related messages. If problem persists, contact IBM Technical Support.

FST0130E ABEND abend-code OCCURRED WHEN OPENING DSN=dsn-name

Explanation: An attempt to open the data set shown in the message failed with an abend.

User Response: Check log for other related messages. Check your JCL and ensure all data sets are allocated and have the correct DCB attributes. Keep any supporting documentation. If problem persists, contact IBM Technical Support.

FST0140E FILE file-name IS NOT ALLOCATED TO THIS SESSION

Explanation: The file shown in the message was not allocated to the session or to the JCL being executed.

User Response: Probable user error. Make sure that file name is allocated to your session. Rerun job. Check your JCL to ensure all files are properly specified.

FST0141E OBTAIN MACRO FAILED FOR FILE file-name (RC=rc-code)

Explanation: Unable to obtain DCB attributes of a file.

User Response: Check log for other related messages. Make sure that all required data sets are allocated, cataloged, and available to the session or JCL. Check your JCL to ensure all files are properly specified.

FST0144E UNABLE TO LOCATE DSNAME INFORMATION FOR dsn-name

Explanation: The data set shown in the message is invalid or is not properly allocated.

User Response: Probable user error. Check JCL or allocations. Make sure all data set names are valid, allocated, cataloged, and available to the application.

FST0145E DEVICE TYPE FOR FILE file-name NOT SUPPORTED

Explanation: The application does not support the device type for the file specified.

User Response: Probable user error. Check the file-name in message. If problem persists, contact IBM Technical Support.

FST0155E RESERVE FAILED WITH RC=rc

Explanation: The application issued a RESERVE macro request, but the RESERVE failed.

User Response: Ensure the database is allocated and has been initialized.

FST0235I text-1 dsn-name

Explanation: This is a tracing message created during dynamic data set allocation.

User Response: none

FST0238E DDNAME=ddname DSN=dsn-name

Explanation: A dynamic allocation error was encountered when processing the file shown in the message.

User Response: Review other log messages. If the data set name is a permanent file, ensure the file has not been deleted or archived. For temporary files, ensure the WORK UNIT and VIO UNIT specification in the INITIALIZATION MODULE file reference a valid UNIT name.

FST0239E DYNAMIC ALLOCATION REQUEST FAILED. RC return-code RE1(reason-code-1) RE2(reason-code-1)

Explanation: This message is issued from \$\$DYNAM services, the dynamic allocation manager. Allocation failed.

User Response: User, setup, or internal error. Check for other messages in the log. Check any IBM messages in the Messages and Codes manual. If unable to resolve, collect all documentation and contact IBM Technical Support.

FST0249E PRIMARY CIGINI MODULE NOT FOUND. INITIALIZATION FAILED

Explanation: The load module CIGINI could not be found in the STEPLIB or link list.

User Response: Ensure the load module CIGINI is located in the STEPLIB, JOBLIB, LINKLIB, or LPALIB. Once the INITIALIZATION MODULE has been placed in the proper load library, retry the job.

FST0250E INVALID RECFM= PARAMETER PASSED TO \$\$DYNAM

Explanation: An internal call was made to dynamically allocate a file, but the file being allocated is of a record format which is not supported.

User Response: Look for other messages in the log, particularly messages which show the data set name being allocated. Ensure that data set name has the correct DCB characters as required by the utility that you are attempting to executed. Specify the correct data set or correct DCB attributes, and restart the job.

FST0251I PRODUCT LOAD LIBRARY.... dsname

Explanation: This message shows the library from which the products software is loaded. This data set name is specified in the primary INITIALIZATION MODULE file.

User Response: none

FST0257I WORK UNIT.....unit-name

Explanation: Displays the work value coded in the primary INITIALIZATION MODULE file.

User Response: none

FST0259I VIO UNIT unit-name

Explanation: This message displays the VIO unit value coded in the primary INITIALIZATION MODULE file.

User Response: none

FST0260E UNABLE TO COMPLETE REPORT SET-UP FOR FILE ddname

Explanation: Failure to open or allocate the ddname shown in the message.

User Response: Look for other messages shown in the log. The most likely cause of this error is that the REPORTER cannot open the report output file. Ensure the report file is a valid QSAM file, and resubmit the job.

FST0261E UNABLE TO ACQUIRE STORAGE FOR REPORT HEADINGS

Explanation: An attempt was made to acquire storage to hold report headings, but storage was not available.

User Response: Increase the REGION parameter, and resubmit the job. If you are requesting reports against very large amounts of data, then selection criteria might be needed to reduce the amount of data processed.

FST0262E FAILURE TO WRITE REPORT HEADING TO FILE ddname

Explanation: An attempt was made to write a report header to the ddname shown in the message, but the write operation failed.

User Response: Ensure the data set name associated with the ddname is a valid QSAM data set. Correct the data set or data set attributes, and resubmit.

FST0270E UNABLE TO ACQUIRE STORAGE DURING PROGRAM LOADING

Explanation: An attempt was made to acquire storage, but the GETMAIN operation failed.

User Response: Increase the REGION parameter, and resubmit the job.

FST0271I PROGRAM LOAD NOTE: //ddname IS NOT ALLOCATED TO SESSION

Explanation: An attempt was made to load a program from the ddname shown in the message, but the load attempt failed.

User Response: none

FST0272E UNABLE TO ALLOCATE DSNAME dataset

Explanation: An attempt was made to dynamically allocate the data set shown in the message, but the allocation failed.

User Response: Ensure the data set exists and is cataloged. Ensure the data set is not in exclusive use by another application. Resubmit the job.

FST0273E UNABLE TO OPEN DSNAME ddname

Explanation: An attempt was made to open the data set shown in the message, but the open failed.

User Response: The data set is most likely not a partitioned data set which contains load modules. Ensure the data set shown in the message is a load library. Correct the error, and resubmit the job. Check your INITIALIZATION MODULE file to ensure the product load library is correct.

**FST0274E UNABLE TO LOAD PROGRAM
program-name FROM dataset**

Explanation: An attempt was made to load the program shown in the message, but the program could not be found.

User Response: The data set is most likely not a partitioned data set which contains load modules. Ensure the data set shown in the message is a load library. If the data set is a load library then ensure the program shown exists in the data set. Correct the error, and resubmit the job. Check your initialization module to ensure the product load library is correct.

**FST0275E FAILURE TO CLOSE DSNAME
dataset-name**

Explanation: An attempt was made to close the load library name shown in the message, but the close was unsuccessful.

User Response: Retry the action. Contact IBM Technical Support if the problem persists.

**FST0276E DDNAME ddname IS ASSIGNED TO
NULLFILE**

Explanation: This is a tracing message which shows a NULLFILE (DD DUMMY) is being associated with the ddname shown in the message.

User Response: Ensure the ddname shown in the message references a data set name rather than a NULLFILE (DD DUMMY).

**FST0277E PROGRAM LOAD WILL NOT BE
PERFORMED**

Explanation: An attempt was made to load a program from a load library, but the load was unsuccessful.

User Response: Ensure the product load library specified in the initialization module references the correct version of the software product you are executing. Look for previous messages in the log for more information. Correct the problem, and resubmit the job.

FST0299I text

Explanation: This message contains descriptive text related to parameters specified in the initialization module.

User Response: NONE

**FST0490E DDNAME ddname IS NOT
ALLOCATED**

Explanation: An attempt to extract directory information was made, but the requested ddname was not allocated to the session.

User Response: Increase the REGION= parameter on the jobcard or step and rerun the job. Contact IBM Technical Support if the problem persists.

**FST0491E dsn-name IS NOT A PARTITIONED
DATA SET**

Explanation: An attempt to extract directory information as made, but the requested data set was not defined as DSORG=PO (partitioned).

User Response: Ensure the data set being processed is a partitioned data set.

FST0492E FAILURE TO OPEN dsn-name

Explanation: An open request was issued against the data set shown in the message, but the open failed.

User Response: Ensure the data set being processed is a partitioned data set. Ensure the data set has not been deleted or archived.

**FST0493E SYNAD FAILURE FOR OPEN FOR
dsn-name**

Explanation: A SYNAD (error handling) exit was taken during open processing for the data set shown in the message.

User Response: Ensure the data set being processed is a partitioned data set. Ensure the data set has not been deleted or archived.

FST0495E UNABLE TO ACQUIRE STORAGE

Explanation: Insufficient storage is available to satisfy request.

User Response: Increase the REGION= parameter on the jobcard or step. Rerun the job. If problem persists, use selection criteria to limit the size of the directory list being processed.

**FST1800E FAILURE TO ALLOCATE SORT
MESSAGE FILE**

Explanation: Unable to dynamically allocate the PGM=SORT message file.

User Response: Look for other messages in the log. There might be dynamic allocation error messages giving clues as to the possible cause of error. Ensure the WORK= and VIO= specification in the primary INITIALIZATION MODULE file point at valid units. Correct the error, and retry the job. If the problem persists, contact IBM Technical Support.

FST1801E SORT ERROR OCCURRED: SORT MESSAGE FOLLOWS

Explanation: An attempt to call PGM= SORT was made, but SORT encountered an error.

User Response: Look at the sort messages that follow this message for clues in problem resolution.

FST1802E END OF SORT MESSAGES

Explanation: An attempt to call PGM= SORT was made, but SORT encountered an error. Look at the previous sort messages for clues in problem resolution.

User Response: Correct the error, and retry the job. If the problem persists, contact IBM Technical Support.

FST1803E SORT WAS UNABLE TO WRITE MESSAGES TO SORT MESSAGE FILE

Explanation: The SORT message file internally allocated by \$\$ SORT might be too small or incorrectly allocated by \$\$ SORT. This error could be an environmental problem specific to your installation.

User Response: Look for other messages in the log. There might be dynamic allocation error messages giving clues as to the possible cause of error. Ensure the WORK= and VIO= specification in the primary INITIALIZATION MODULE file point at valid units. Correct the error, and retry the job. If the problem persists, contact IBM Technical Support.

FST1804E UNSUCCESSFUL EXTERNAL SORT: RC= return code

Explanation: This message follows previous SORT error messages.

User Response: Look for other messages in the log. Correct the error, and retry the job.

FST1805E UNSUCCESSFUL INTERNAL SORT: RC= return code

Explanation: This message follows previous SORT error messages.

User Response: Look for other messages in the log. Correct the error, and retry the job. You might need to increase the REGION= parameter.

FST1806E FAILURE TO READ SORT MESSAGE FILE. RC= return code

Explanation: A read error occurred by program \$\$ SORT when attempting to read the SORT message file.

User Response: Look for other messages in the log. Correct the error, and retry the job. If error persists, contact IBM Technical Support.

FST1807E FAILURE TO WRITE SORT MESSAGE TO MESSAGE QUEUE. RC=return code

Explanation: Most probable cause is insufficient REGION available to internally called SORT message routines.

User Response: Look for other messages in the log. Increase your REGION size. Correct the error, and retry the job.

FST1808E GETMAIN FAILURE FOR PROGRAM STACK IN \$\$ SORT2

Explanation: Most probable cause is insufficient REGION. Look for other messages in the log.

User Response: Increase your REGION size. Correct the error, and retry the job.

FST1809I STARTING INTERNAL SORT: RECORD COUNT= count

Explanation: The message is issued when \$\$ SORT is called and // CIGTRACE is allocated to NULLFILE.

User Response: none

FST1810I STARTING EXTERNAL SORT: RECORD COUNT= count

Explanation: The message is issued when \$\$ SORT is called and // CIGTRACE is allocated to NULLFILE.

FST1811I FINISHED WITH SORT

Explanation: The message is issued when \$\$ SORT completes calling either the internal or external sort.

User Response: none

FST1812I USING EXTERNAL SORTWORK FILES

Explanation: The trace message indicates that the external SORT program will be invoked.

User Response: none

FST1813I USING "DYNAM" OPTION WHEN CALLING EXTERNAL SORT

Explanation: The trace message indicates that the DYNAM option will be specified when invoking the external SORT program will be invoked.

User Response: To avoid using the DYNAM option, you can pre-allocate SORTWKxx files to the job and resubmit.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue

Notices

San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

Advanced Function Printing	MVS
AFP	MVS/DFP
BookManager	MVS/ESA
CICS	OS/390
DB2	RACF
DFSMS	z/OS
DFSMS/MVS	

Other company, product, and service names may be trademarks or service marks of others.

Bibliography

Merge Tool publications

License Information, GI10-8473

User's Guide, SC27-1694

Program Directory, GI10-8464

Related publications for OS/390

OS/390 ISPF User's Guide, SC28-1239

OS/390 MVS JCL Reference, GC28-1757

OS/390 MVS System Management Facilities (SMF),
GC28-1783

OS/390 TSO/E REXX User's Guide, SC28-1974

OS/390 TSO/E REXX Reference, SC28-1975

OS/390 DFSMS Access Method Services for Catalogs,
SC26-7326

OS/390 DFSMS: Using Data Sets, SC26-7339

TSO/E Command Reference, SC28-1881

TSO/E Programming Services, SC28-1875

Related publications for z/OS

z/OS ISPF User's Guide Vol 1, SC34-4822

z/OS ISPF User's Guide Vol 2, SC34-4823

z/OS MVS JCL Reference, SA22-7597

z/OS TSO/E REXX User's Guide, SA22-7791

z/OS TSO/E REXX Reference, SA22-7790

z/OS DFSMS Access Method Services for Catalogs,
SC26-7394

z/OS DFSMS: Using Data Sets, SC26-7410

z/OS TSO/E Command Reference, SA22-7782

z/OS TSO/E Programming Services, SA22-7789

Softcopy publications for Merge Tool

The following collection kits contain Merge Tool or related product publications in BookManager[®] format:

OS/390 Collection, SK2T-6700

Index

A

actions
 analysis 4
 batch utility 3
 browse 65
 complex 4
 conflict 4
 edit member 65
 edit work file (EW) 71, 74
 EW 68, 71, 74
 exclude 26
 exclude member 65
 existing release 3
 GB 51, 74
 generate merge file 51
 generate merge file from work file 65, 66
 generate statistics 66
 generate work and merge file 65, 66
 generate work file 65, 66
 generate work file (GW) 74
 GM 51, 68, 69
 group 71, 73
 GW 74
 include excluded member 65
 ISPF 67
 maintenance 3
 programmer 71, 73
 promoted 72
 reconcile 3
 retrofit 4
 show statistics 65
 submitted 57, 59
 work file 3
Actions column
 clear 25, 66
allocate data set 14, 15
application maintenance 67
asm 63
ASM 8
AZZCBL01 63
AZZFRS02 75
AZZISPF 75
AZZMERG0 60
AZZMRKUP 59
AZZSAM1 75
AZZSAM2 75
AZZSAMB 75
AZZSAML 75
AZZSAMB 75
AZZSAMS 75
AZZSAMW 75
AZZSAMZ 75
AZZV1R0 75

B

base
 file 21

base file
 AZZCBL prefix 24
 base 73
 DD 55
 matched member 21
 name pattern 63
 original 5, 71, 73
 source 71
 vendor source 73
 version1 5
 version2 5
batch 4
batch jcl 70, 73
batch processing
 prerequisite knowledge v

C

CIGIN 5
CIGSTATS 5
CIGSUMM 5
CLEAR command 25, 66
cobol 63
COBOL 8, 15
column
 clear 25, 66
 end 21
 start 21
command
 set 65, 66
commands
 CLEAR 25, 66
 eXclude 25
 L 25, 66
 LOCATE 25, 66
 primary 24
 RES 25, 66
 RESET 25, 66
 X [ALL] 66
 X ALL 25
compare columns 72
components 6
configure 70
conflicting 2
consultants 72

D

data 63
DATA 15
data set 75
data set allocate 14, 15
data sets 67
DD
 base file 55
 cigbase 55, 59, 60, 61
 cigdd01 55, 59, 60, 61, 70, 73
 cigdd02 55, 59, 60, 61, 70, 73
 cigin 55, 62
 CIGIN 5

DD (*continued*)

 ciglog 55
 cigmerge 55, 59, 61
 cigstats 55
 CIGSTATS 5
 cigsumm 55
 CIGSUMM 5
 cigwork 59, 61, 70, 73
 file 2 55
 merge file 55
 statements 55
 STEPLIB 75
 sysout 55
 SYSOUT 5, 15
 sysproc 75, 76
 work file 55

development
 assess effort 4
 basic parallel 1
 common parallel 1
 release management 1
 scenarios 1
 vendor application 1
development effort 72
DISP: mod 12

E

example
 application 67
 example1 63
 example2 63
 example3 63
Exclude All command 25, 66
excluded 26
Exec Mode 8

F

field
 append to list 21
 line command 20
file 1
 customization 74
 customized 73
 DD 55
 name pattern 63
 production 70, 71
 source 71
file 2
 AZZCBL prefix 24
 changes 74
 DD 55
 functionality 67
 inserts 8
 member 20
 name pattern 63
 new release 70, 71, 73, 74
 new version 73
 required changes 70

- file 2 (*continued*)
 - retrofit 73
 - second fix 69
 - upgrade 73
 - version 73
- file1
 - inserts 8
 - member 20
 - member list 20
 - member name 21
- File1
 - compared 5
 - merged 5
- file1 file
 - AZZCBBY prefix 24
- files
 - work file 4
- forecast 72
- foreground 4, 14

G

- Group Action 8

H

- help facility 9
- high level qualifier 59
- hlq 59

I

- ISPF
 - actions 67
 - help facility 9
 - libraries 67
 - library 70, 72
 - main panel 76
 - panel 75, 76
 - panel changes 76
 - primary panel 77
 - skeleton library 75
 - workbench 2, 6, 9

J

- jcl 63
- JCL 8, 15
 - generated 58
 - submitted 57, 59

L

- library
 - clist 75, 76
 - data sets 70, 73
 - file1 20
 - file2 member 20
 - input 13
 - ISPF 72
 - link list 75, 76
 - lla 76
 - load library 75
 - mark-up request 35
 - member list 20

- library (*continued*)
 - merge file 21
 - output 14
 - review location 73
 - SAZZCLIB 75, 76
 - SAZZLOAD 75
 - search order 13
 - set panel 12
 - settings 7
 - SISPPENU 76
 - skeleton library 75
 - SYS1.LOCAL.ISPFNLS 76
 - SYS1.PARMLIB 76
 - SYS1.TSOPROC. 75
 - update 71, 73
 - work file 21
- life cycle 68
- lifecycle 69
- line command
 - field 20
- lla 76
- load library 75
- LOCATE command 25, 66
- log file
 - error 5
- Logon Procedure 75

M

- mark up and analyze 6
- mark up indicators 5
- marked-up
 - work file 2
- member
 - actual name 17
 - append 21, 72
 - blank 20
 - browse 65
 - build list 70, 73
 - compare columns 72
 - compiled 53
 - consolidate 71
 - consolidation 73
 - corresponding 21
 - create list 21
 - edit 65
 - exclude 26
 - exclude member 65
 - include excluded member 65
 - linked 53
 - list 8
 - locate 25, 66
 - missing 21
 - non-excluded 25
 - omit 71
 - omitting 70, 73
 - patterns 23
 - prefix 20
 - prog00 76
 - query 21
 - same names 18
 - sample JCL 59
 - select name 67
 - specify name 67
- member list
 - primary commands 24
 - remove 25, 66

- member statistics
 - sample report 31
- members
 - match up 6
- merge
 - automatic 2
 - coexist 74
 - compare columns 63
 - conflict 39, 68, 74
 - conflicting 2
 - consolidate 71
 - consolidation 68
 - create source 6
 - default 63
 - directives 6, 53, 69, 71, 74
 - entire library 21
 - GM 69
 - included 71
 - library 21
 - marked-up file 42
 - member list 19
 - objects 62
 - output 53, 71
 - outputs 53
 - overlap 39
 - parameters 15
 - process 5
 - review 6
 - same file name 53
 - scope 67
 - statistics 2
 - syntax 62
 - tasks 71, 73
 - test output 6
 - versions 2
- merge file
 - add source 74
 - coexist 74
 - conflict 74
 - consolidation 68
 - DD 55
 - directives 68, 69
 - generate merge 74
 - generate merge (GM) 71
 - GM 71, 74
 - mark up indicators 5
 - merged output 5
 - output 5
 - outputs 74
 - review 74
- merge process 2
- Merge Tool
 - functions, overview 1
- message
 - detailed 9
 - long 9
- mod 12
- modifications
 - conflicting 2
- module 13

N

- notices 85

O

- online ordering 67
- option default settings 18
- optional
 - report files 14
- overlap area
 - location 5
 - work file 5
- overview
 - Merge Tool functions 1

P

- panel
 - action 7
 - allocate 14, 15
 - append list 7
 - append to list 21
 - build final merge file 61
 - build list 19
 - build mark-up file 60
 - changes 76
 - command line 9
 - conflicting source options 46, 47, 49
 - create merged file 62
 - edit and submit JCL 57
 - fields 7
 - group action 26
 - have merge column 52
 - have work column 38, 52
 - indicator 24
 - information 7
 - ISR@PRIM 76
 - main 7, 67, 76
 - member 7
 - member actions 9
 - member exclude 26
 - member list 20
 - member match 21
 - member prefix 20
 - message 9
 - option default 18
 - pattern supplied 24
 - pop-up 12, 21
 - preferences 23
 - primary 77
 - progress indicator 37, 52
 - query 19
 - resulting member list 24
 - search order 13
 - session 7
 - settings 18
 - specify member 8, 23
 - specify members 24
 - specify source type 16
 - tutorial 9
 - valid commands 65
- parallel development 1
- parameter
 - optional 63
 - required 63
 - select 62
- partitioned data set 75
- pattern supplied 24
- pds 4
- PF1 9

- PL1 15, 63
- preference
 - session 67
- primary commands 24
- processing options
 - global
 - setting for Merge Tool 1
- production
 - fix 3
 - missing fix 3
 - source 3
- production fix 68
- programmer 68
 - initials 17
- project management 70
 - analysis 2
- promote 69

R

- release
 - existing 3
 - maintenance 3
- report
 - analyze 71, 73
 - build work process 29
 - conflicting changes 31
 - lines deleted 31
 - lines inserted 31
 - log activity 35
 - mark-up summary 29
 - member statistics 31
 - review 69, 71
 - select 70, 73
 - summary statistics 33
 - types 29
 - Work File statistics 31, 33
- reports
 - statistics 2
- RES command 25, 66
- RESET command 25, 66
- review 6

S

- sample
 - application 75
 - AZZBLD1 75
 - AZZCBL01 63
 - AZZFRS02 75
 - AZZISPF 75
 - AZZMERG0 60
 - AZZMRGIT 61
 - AZZMRKUP 59
 - AZZSAM1 75
 - AZZSAM2 75
 - AZZSAMB 75
 - AZZSAML 75
 - AZZSAMM 75
 - AZZSAMS 75
 - AZZSAMW 75
 - AZZSAMZ 75
 - AZZV1R0 75
- SAZZCLIB 75, 76
- SAZZLOAD 75

- scenario
 - sample 67
 - scenario1 67
 - scenario2 68
 - scenario3 70
 - scenario4 72
 - usage 67
- scenarios 1
- SCLM 69
- search order 13
- sequential data set 75
- session
 - configure 73
 - preference 73
 - preferences 69, 71, 73
- Settings (option 0) 1
 - description 1
- SISPPENU 76
- SMP/E 75
- source
 - applied 70
 - asm 63
 - ASM 8
 - base 70, 73
 - base source 69
 - both versions 71
 - cobol 63, 70, 73
 - COBOL 8, 15
 - columns 63
 - common version 67
 - complexity 71
 - conflicting 5
 - copybooks 70, 73
 - customization 72
 - customized 73
 - data 63
 - DATA 15
 - delete 74
 - deletes 69, 71, 73
 - development effort 72
 - different 5
 - enhancement 67
 - incorporate 68
 - inserts 69, 71, 73, 74
 - jcl 63, 70, 73
 - JCL 8, 15
 - libraries 72
 - life cycle 74
 - management tool 68, 69
 - merge 63
 - merge file 72
 - modifications 29
 - modify versions 11
 - new release 70
 - original 67, 70, 73
 - original source 69
 - other fix 69
 - PL1 15, 63
 - previous version 73
 - proc 70, 73
 - production 70, 73
 - production fix 67, 69
 - promoted 69
 - promotion 70
 - reconcile 72
 - reconciliation 70
 - second fix 69

- source *(continued)*
 - type 8
 - unload 72
 - vendor 72, 73
 - vendor source 72
 - version 68
- statistics
 - member 15
 - summary 15
- statistics file
 - CIGSTATS 5
 - CIGSUMM 5
 - writes 5
- steps
 - analyze and mark up 5
 - construct work file members 29
 - create source 5
 - customization 75
 - GM action 68
 - identify 5
 - ivp 75
 - marked-up work file 67
 - over and over 53
 - repeated 69, 72
 - review changes, address conflicts 5
 - session 69
 - skipping 71
 - test output 5
- summary statistics
 - sample report 33
- syntax
 - name pattern 63
- SYS1.LOCAL.ISFPNLS 76
- SYS1.PARMLIB 76
- sysout 55
- SYSOUT 5

T

- terms
 - conflict 5
- tools 2
- TSO 73
 - clist 75, 76
 - link list 76
 - lnkfst 76
 - logon procedure 75
 - rdefine 75
 - tsoproc 75
- tsoproc 75

V

- View Reports 8

W

- work file
 - addressing conflict 51
 - allocation 4
 - combined changes 51
 - complexity 74
 - conflicting changes 68
 - conflicts 6
 - consolidate 69
 - DD 55

- work file *(continued)*
 - deletes 68, 71
 - directives 68, 71, 74
 - document 6
 - edit 68, 69, 74
 - edit work file (EW) 74
 - edited 72
 - final merge 51
 - generate 67, 69, 71
 - Generate Merge file 69
 - GM 69
 - GW 71
 - inserts 68, 71
 - intermediate 4
 - marked-up 14, 70, 71, 73
 - marked-up file 69
 - merged 68, 72
 - optionally modified 51
 - partitioned data set 14
 - repeat 74
 - retrofit 74
 - review 51, 68, 74
 - tests fail 68
- workbench 2

X

- X ALL command 25, 66

Readers' Comments — We'd Like to Hear from You

IBM Merge Tool for z/OS and OS/390
User's Guide
Release 1

Publication No. SC27-1694-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
H150/090
555 Bailey Avenue
San Jose, CA
U.S.A.
95141-9989



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5697-H70

Printed in USA

IBM Merge Tool for z/OS and OS/390 publications

GI10-8473	License Information
GI10-8464	Program Directory

SC27-1694-02

