Chicago Interface Group, Inc.

CIG Merge Tool

Reference Guide

Chicago Interface Group, Inc.
858 West Armitage Avenue #286
Chicago, IL 60614   USA

Phone: (773) 524-0998
Fax: (815) 550-6088
Internet: www.cigi.net
Email: cigi_sales@www.cigi.net

TABLE OF CONTENTS

CHAPTER 1: Product overview.......... 6

Note: CONFLICTING modifications may require manual intervention.......... 7

CHAPTER 2: The CIG Merge Tool ISPF Workbench   12

See Column Headings in the previous section for more detail about each column.............................. 19

When the action is SS the result will first be a foreground display of the member statistics................... 28

The types of reports are described in the following table................. 29

CHAPTER 3: The CIG Merge Tool steps. 31

A simple example of conflicting source changes follows..................... 33

CHAPTER 4: The CIG Merge Tool Batch Utility   35

CHAPTER 5: CIG Merge Tool Reports and Statistics 40

Appendix A – Sample CIG Merge Tool Scenarios............................ 43

CHAPTER 1: PRODUCT OVERVIEW

Parallel Development

Definition

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 will find themselves performing tasks against the same code in parallel with other programmers.

Parallel Development Basic Steps

The basic steps for parallel development follow. These steps are common in most parallel development scenarios.

➢ Identify what components have changed
➢ Analyze the complexity of the project
➢ Merge and review the changes
➢ Address conflicting changes
➢ Create merged source
➢ Test outputs from merged source

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 is the CIG Merge Tool for OS/390 and z/OS?

Overview

The CIG Merge Tool is a source code merging utility that allows two versions of source code to be automatically merged together. The merge process 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, moved, and conflicting lines.
2. A merged file member containing all modifications to the source module.
3. Statistics showing the number of modifications broken out by inserted, deleted and shifted lines. The statistics also show the number of conflicting modifications.

Note: CONFLICTING modifications may require manual intervention.

The CIG Merge Tool will enable 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.

Functionality

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

➢ 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

The CIG 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 will change is who will use the tools and perform 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 CIG Merge Tool actions and reports in foreground or batch through a single ISPF panel.

➢ Reports: The CIG Merge Tool provides reports that will 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 CIG 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 will take place. This is the sole input to the Merge action.

Parallel Development Solutions with the CIG Merge Tool

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

Common parallel development: Production fix is coded to same program being upgraded for an enhancement.

Problem: The source for the enhancement does not contain the production fix. Promotion of the enhancement will reverse 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 the CIG 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 will cause source regression and production failures to return.

Solution: Programmers must incorporate the independently coded fixes, using the CIG 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 will reverse 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 the CIG 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.

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, it should not be necessary to entirely re-code the customization.

Solution: Using the CIG 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 CIG Merge Tool

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

The Work File

What is a Work File?

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

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

Allocation

The 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 the user is prompted with the file allocation panel if the defined file does not exist. For batch execution, this file is required if you wish to create it or if you wish to create a merged file.

The Base File

What is a Base File?

The Base File is the PDS that contains the original version of source common to both versions of source. Both Version1 and Version2 should have used the source in this library as the base or starting point for their changes. This file is required if you wish to create a work file.

The File1

What is File1?

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

The File2

What is File2?

File2 is the second of two versions of source to be compared and merged with the Base File. This file is required if you wish to create a work file.

The Merge 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 Source Management Tool creating output that can be tested for the functions of both File1 and File2 modifications. This file is required for the final merge process.

The Log File

What is the Log File?

The 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. Additionally, if any tracing parameters are passed via the CIGIN DD statement the resulting trace data is written to this file.

This file may be a sequential dataset, SYSOUT, or a partitioned dataset.

The Statistics Files

What are the Statistics Files?

The 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 the CIG Merge Tool works

The Merge Process

As stated earlier the basic steps for parallel development are:
1. Identify what components have changed
2. Analyze the complexity of the project
3. Merge and review the changes
4. Address conflicting changes
5. Create merged source
6. Test outputs from merged source

All but one of these steps, 6.Test outputs, is can be performed directly from within the CIG Merge Tool. The steps are described below.

1. Identify what components have changed. Working with the Merge Tool the user will match up the Base File members with members of the same name or pattern from File1 and File 2. Based on the match up of members the components that require merging can be easily identified.
2. Analyze the complexity of the project. Using the Merge tool the user can request a preliminary building of the Work File to obtain member statistics such as number of inserts, deletes and conflicting changes. Based on the statistics, the user can get a better understanding of the resources required for the merge process.
3. Merge and review the changes. The Merge Tool is invoked to build a marked-up work file by comparing the base program and one or two files that originated from the base file. The Merge Tool places merge directives in the first seven bytes of the Work File to document the insertions, deletions, and overlap areas. Merge directives are used to determine what action the merge step will take for that line of source.
4. Address conflicting changes. Via the Merge Tool the programmer can edit the Work File to resolve conflicting change issues. The programmer can try different scenarios by changing the merge directives.
5. Create merged source. In this step the Merge Tool is used to merge the Work File into a Merge File. The Merge File is source that can be stored in the site Source Management Tool, or used as input to a compiler. The Work File is the sole input to the merge step. The Merge Tool reads the input and performs the insertions and deletions indicated in the first seven bytes of the Work File.
6. Test Outputs from merged source.   The final step is to add the merged source into SCLM creating outputs that can be tested for all functionality. If the testing is unsuccessful, the user can perform steps 3 – 6 repeatedly until successful testing is achieved.   At this time the Merge Tool does not interface with or automatically add the merged source into any source management tool.

Merge Tool Usage Notes

The member list is driven off the File1 member list. Only members that match the member names in File1 or the pattern supplied in the Specify Members panel will be displayed in the list.

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

CHAPTER 2:   THE CIG MERGE TOOL ISPF WORKBENCH

The CIG Merge Tool ISPF panels enable the user to work with the CIG Merge Tool interactively. You can work with all or some members that exist in the libraries with which you are working.

The CIG Merge Tool Main Panel

```
                              CIG Merge Tool
 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
 ****************************** Bottom of data ********************************
```

The main panel is broken into three sections: Session information, Action information and the member list section.

The panel fields are described in the following tables.

Panel Fields

Field	Description
Member	Specify a member name or a member pattern to work with specific members or leave blank for all members. You can plan a "/" in the member field to access the Specify Members panel.
Append to list	Indicates if 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	Specify libraries or change your current libraries settings.
Source Type	Source type is used by the CIG 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.
Group Action	Enter "/" to see group action choices. A group action will apply 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.

Line Commands

Enter "/" (across from a member) to see line command choices. A line command will apply only to the member on that line.

```
                          CIG Merge Tool                    Row 1 to 8 of 8
 C .---------------- Merge Tool Member Actions ----------------. ll ===> PAGE
   | Command ===> _____     | ect option.
 M |                                                            | _ Preferences
 - |   Member Action for member  AZZCBLA1                       | ------------
 _ |                                                            | und 2. Batch)
   |   __ 1.  GW-Generate Work File                             |
 " |      2.  GB-Generate Work and Merge File                   |
 / |      3.  GM-Generate Merge File from Work File             |
 _ |      4.  SS-Generate Work and 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                       |
 _ |                                                            |
 * |   Select a choice and press ENTER to process member action. | *************
   '------------------------------------------------------------'
```

Column Headings for member lists

Heading	Description
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 will show "YES". If a corresponding member does not exist, it will show "-------". If members are not named the same and a corresponding member exists, it will show the corresponding member name. If a corresponding member does not exist, it will show "-------".
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 will show "YES". If a corresponding member does not exist, it will show "-------". If members are not named the same and a corresponding member exists, it will show the corresponding member name. If a corresponding member does not exist, it will show "-------".
Have Work	An entry indicates whether or not a marked-up file exists in the library specified as the work file. The 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. The 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 will be 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.

Preferences

The CIG Merge Tool preferences allow you to set preferences that will be in effect while you use the CIG Merge Tool.

```
                        Merge Tool Option Default Settings
OPTION  ==>


This panel displays the current setting for each of the Merge Tool options.
Users can modify values on this panel.


Panel Options:
 View or Browse           ==>   (V or B)
 Member Names the Same   ==>   (Y or N)
```

Preference Panel Options

Option	Description
View or Browse	Enter "V" if you want to use ISPF VIEW when looking at files or "B" if you want to use ISPF BROWSE. The default is "B".
Member Names the Same	If the member names that correspond to each other are not exact matches, enter "N" to indicate that member names are not the same.

Note: When you specify the same library for File1, File2, or Base, the CIG Merge Tool knows that the members' name cannot be the same. Conversely, if you have specified three unique libraries, the CIG Merge Tool assumes that your member names will match. You override this assumption by specifying "N" after "Member Names the Same".

Source Type

Enter "/" to specify or change the current source type setting (COBOL, ASM, JCL, DATA, PL1). Source type is used by the CIG Merge Tool to determine where line numbers are, so that they can be ignored. Additionally, you can specify the actual start and end columns to be compared.

```
                             CIG Merge Tool                    Row 1 to 8 of 8
C .------------------- Specify Source Type -------------------. ll ===> PAGE
  |                                                            | ect option.
M | Command ===> _____ | _ Preferences
- |                                                            | -----------
_ |    Specify Type:   _____  (COBOL, ASM, JCL, DATA, PL1)   | und 2. Batch)
  |                                                            |
" |                                                            |
_ |                                                            |
_ |        or                                                  |
_ |                                                            |
_ |   Specify compare column start  ____   and end  ____       |
_ |                                                            |
_ |                                                            |
_ |                                                            |
* |                                                            | *************
  '------------------------------------------------------------'
  '
```

Set Libraries

The Set Libraries pop-up panel is used to specify the partitioned data sets that The user will use for the merge process.

```
Merge Tool
 C .-------------------- Merge Tool Set Libraries --------------------. PAGE
   | Command ===>                                                     | ion.
 M |                                                                  | rences
 - |               Input Libraries (Required)                         | -----
 _ |  File1          'AZZ.V0R1.SAZZSAM1'                               | Batch)
   |  File2          'AZZ.V0R1.SAZZSAM2'                               |
 " |  Base           'AZZ.V1R0.SAZZSAMB'                               |
 * |                                                                  | ******
   |                                                                  |
   |                                                                  |
   |                                                                  |
   |               Output Libraries (Required)                        |
   |  Work           'AZZ.V0R1.SAZZSAMW'                               |
   |  Merge          'AZZ.V1R0.SAZZSAMM'                               |
   |                                                                  |
   |               Output Report Files (Optional)                     |
   |  Member Stats   'AZZ.V0R1.SAZZSAMS'                               |
   |                 Disp:    mod                                      |
   |  Summary Stats  'AZZ.V0R1.SAZZSAMZ'                               |
   |                 Disp:    mod                                      |
   |  Log            'AZZ.V0R1.SAZZSAML'                               |
   |                 Disp:    mod                                      |
   |                                                                  |
   '------------------------------------------------------------------'
```

Input Libraries

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

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.
Base	Specify the name of the library that contains the original source modules that has the basis for the source modules contained in "File1" and "File2". Up to four libraries can be specified. The libraries will be searched in the order as specified on the panel.
Work	Specify the name of a partitioned data set. If it does not exist, you can let the CIG Merge Tool automatically allocate it. You will be prompted, if it does not exist.

Output Libraries

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

Library	Description
Work	Specify the name of the partitioned data set. This is where the intermediate merged file members reside.
Merge	Specify the name of a partitioned data set. If it does not exist, you can let the CIG Merge Tool automatically allocate it. You will be prompted, if it does not exist.

Output Report Files

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

When you execute the CIG Merge Tool in foreground, you will need to specify files in order to view the member statistics and the summary statistics.

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

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 will probably need to manually intervene making sure the merged source module will function 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 the file you specify does not exist, the user can let the CIG Merge Tool automatically allocate it.

Below is an example of the Allocate panel the user will be prompted with if a report dataset does not exist.

```
                      CIG Merge Tool for OS/390 and z/OS
C .------------------ Allocate Dataset for Report ------------------.  PAGE
  | Command ===> _____           | ion.
M |                                                                | rences
- |    File for CIGLOG  report does not exist:                      | -----
_ |                                                                | Batch)
  |         'AZZ.V1R0.SAZZSAML'                                     |
" |                                                                |
* |    Optional, specify space:                                    | ******
  |       Tracks _ or Cylinders _                                  |
  |       Primary Space _____   Secondary Space _____            |
  |                                                                |
  |    Optional, select another volume:                            |
  |       Volume _____                                            |
  |                                                                |
  |    To automatically allocate it, Press Enter or PF3 to Cancel  |
  |                                                                |
  |                                                                |
  '----------------------------------------------------------------'
```

Below is an example of the Allocate panel the user will be prompted with if a merge dataset does not exist.

```
                        CIG Merge Tool
C .----------------------- Allocate Library -----------------------.  PAGE
  | Command ===> _____         | ion.
M |                                                                | rences
- |    File for MERGE library does not exist:                       | -----
_ |                                                                | Batch)
  |         'AZZ.V1R0.SAZZSAMM'                                     |
" |                                                                |
* |    To allocate it with same characteristics of                 | ******
  |                                                                |
  |         'AZZ.V1R0.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.               |
  |                                                                |
  '----------------------------------------------------------------'
```

Error messages.

The CIG 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.

```
                          CIG Merge Tool                          Row 1 to 8 of 8
C  .------------------ Merge Tool Member Actions ----------------. ll ===> PAGE
   | Command ===> _____     Invalid value | ect option.
M  |                                                 | _ Preferences
-  |  Member Action for member  AZZCBL01             | ------------
_  |                                                 | und 2. Batch)
   |  __  1.  GW-Generate Work File                  |
"  .--------------------------------. File           |
/  | Enter one of the listed values. |  Work File    |
_  '--------------------------------' tatistics 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             |
   |                                                 |
   |  Select a choice and press ENTER to process member action.  |
   '------------------------------------------------------------'
```

Merge Tool ISPF 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.

18

Building member lists

Once the preferences and libraries have been set for the query, the user is ready to build a list of members for the merge process.

If there is a specific member that the user is working on they should enter the entire member name in the member field of the main panel. Additionally, if there is a common prefix to the member names a name mask can be used (i.e. ABC* will request a list of members whose names begin with ABC). For full member lists and matching the member name should remain blank.

Based on the information on the Merge Tool main panel and from the Libraries panel a list of members is driven when the user hits enter.

```
                          CIG Merge Tool                        Row 1 to 3 of 3
Command ===> _____      Scroll ===> PAGE
                                                         Enter "/" to select option.
Member: AZZCB*___   _ 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
__  AZZCBL01            YES      YES                    _____
__  AZZCBL02            YES      ------                 _____
__  AZZCBY01            ------   YES                    _____
***************************** Bottom of data *******************************
```

See Column Headings in the previous section for more detail about each column.

When attempting to merge entire libraries the Merge Tool assists in identifying what libraries have a matched member to the Base File at a glance.

The member list is driven off the File1 member list. Only members that match the member names in File1 or the pattern supplied in the Specify Members panel will be displayed in the list.

```
                          CIG Merge Tool                        Row 1 to 8 of 8
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
__  AZZASM01            YES      YES                    _____
__  AZZCBLA1            YES      YES      YES           _____
__  AZZCBLA2            YES      YES                    _____
__  AZZCBL01            YES      YES      YES   YES     _____
__  AZZCBL02            YES      ------                 _____
__  AZZCBY01            ------   YES                    _____
__  AZZEOFCD            ------   ------                 _____
__  AZZSWTCH            ------   ------                 _____
***************************** Bottom of data *******************************
```

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

Member lists when the member names are different

If the user has specified on Preferences Panel the member names in the libraries then the names may not be the same; the user will be prompted with the Specify Members panel. This is the panel where the user can specify a member name mask. The Merge Tool will use the patterns to determine if there are any member names that match.

The panel below is telling the Merge Tool that the members in File2 and the Base file will begin with AZZCBL and the members in File1 will be prefixed with AZZCBY, perhaps the programmer's initial.

```
                         CIG Merge Tool
C .--------------------- Specify Members ---------------------. ll ===> PAGE
  |                                                           | ect option.
M | Command ===> _____       | _ Preferences
- |                                                           | ------------
_ |     File1 Member:  AZZCBY*____                            |  und 2. Batch)
  |     File2 Member:  AZZCBL*____                            |
" |     Base Member:   AZZCBL*____                            |
* |                                                           | *************
  |                                                           |
  |                                                           |
  |   Please specify member name or pattern for User2 library |
  |   and your Base Library.  Leave it blank to use same name |
  |   as File1 member.                                        |
  |                                                           |
  |                 Current Libraries                         |
  |                                                           |
  |   File1:  'AZZ.V1R0.SAZZSAM1'                             |
  |   File2:  'AZZ.V1R0.SAZZSAM2'                             |
  |   Base:   'AZZ.V1R0.SAZZSAMB'                             |
  |                                                           |
  |                                                           |
  '-----------------------------------------------------------'
```

The Merge Tool will match the member names based on the pattern supplied and then display the resulting member list.

```
                         CIG Merge Tool                    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                 _____
***************************** Bottom of data *****************************
```

Note that the 'different member name' replaces the YES indicator, informing you that there is a matched member with a different name.

Interpreting the member list

In the following example there are three members that have a File 1, File2 and Base member match.

```
                        CIG Merge Tool                      Row 1 to 5 of 5
Command ===> _____  Scroll ===> PAGE
                                              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                    _____
__  AZZCBLA2          YES      YES                    _____
__  AZZCBL01          YES      YES                    _____
__  AZZCBL02          YES      ------                 _____
__  AZZCBY01          ------   YES                    _____
***************************** Bottom of data *****************************
```

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

Appending to the member list

When the user changes the content of the member field and presses enter the Merge Tool assumes that a new list is being driven. The Merge Tool will replace the contents of the current member list with the results of the new query.

When the user wants to create a new member list but wishes to keep the current list he must request that the Merge Tool append the new results to the current list. This is done via the "Append to list" field.

In the following example the user has requested that any members that begin with AZZCBLA* from these files be added to the already generated list. Note that the two members AZZCBLA1 and AZZCBLA2 have been added to the previous member list.

```
                        CIG Merge Tool                      Row 1 to 6 of 6
Command ===> _____  Scroll ===> PAGE
                                              Enter "/" to select option.
Member: AZZCBLA*  / 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                    _____
__  AZZCBLA2          YES      YES                    _____
__  AZZCBL01          YES      YES                    _____
__  AZZCBL02          YES      ------                 _____
__  AZZCBY01          ------   YES                    _____
***************************** Bottom of data *****************************
```

Merge Tool Actions

Member Actions

The CIG Merge Tool uses a set of line commands or actions that can be executed against any member within the member list.   The action is executed for that member only.   Below is a list of actions and their descriptions.

Action	Description	Notes
GW	Generate Work File	
GB	Generate Work and Merge File	
GM	Generate Merge File from Work File	
SS	Generate Work and Show Statistics for Member	
E (EB, E1, E2, EW, EM)	Edit Member	ISPF only (see note
B (BB, B1, B2, BW, BM)	Browse Member	ISPF only (see note
X	Exclude Member	ISPF only
I	Include Excluded Member	ISPF only Includes excluded

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. The library selection list will be based on the entries defined in the Set Libraries panel.

```
                        CIG Merge Tool                        Row 1 to 8 of 8
C .------------------------------------------------------------. ll ===> PAGE
  | Command ===> _____     | ect option.
M |                                                            | _ Preferences
- |    Select desired library: _   Member: AZZCBLA2            | ------------
_ |                                                            | und 2. Batch)
  |    1. File1  'AZZ.V1R0.SAZZSAM1'                           |
" |    2. File2  'AZZ.V1R0.SAZZSAM2'                           |
_ |    3. Base   'AZZ.V1R0.SAZZSAMB'                           |
E |    4. Work   'AZZ.V1R0.SAZZSAMW'                           |
_ |    5. Merge  'AZZ.V1R0.SAZZSAMM'                           |
_ |                                                            |
_ |  Select library and press ENTER to  EDIT                   |
_ |                                                            |
_ |                                                            |
_ |                                                            |
* '------------------------------------------------------------' *************
```

This panel can be bypassed by entering the extended Edit or Browse action on the main panel (EB, BB, E1, BW etc.).

To generate a list a valid commands the user can enter "/" next to the member. A Prompt panel will be returned.

```
                          CIG Merge Tool                       Row 1 to 8 of 8
C .---------------- Merge Tool Member Actions ----------------. ll ===> PAGE
  | Command ===> _____ | ect option.
M |                                                           | _ Preferences
- |   Member Action for member   AZZCBLA2                     | ------------
_ |                                                           | und 2. Batch)
  |   __ 1.  GW-Generate Work File                            |
" |      2.  GB-Generate Work and Merge File                  |
_ |      3.  GM-Generate Merge File from Work File            |
/ |      4.  SS-Generate Work and 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                      |
_ |                                                           |
_ |   Select a choice and press ENTER to process member action. |
* '-----------------------------------------------------------' *************
```

The user must enter the number (1 through 8) for the corresponding action they wish to invoke.

Include eXclude Actions

The exclude action (X) will virtually remove the member from any group action processing performed against the list. The "EXCLUDED" note is placed to the far right of the member who has been excluded. The note will remain in this place until the list is rebuilt or the member is the object of the include (I) action.

```
                          CIG Merge Tool                       Row 1 to 8 of 8
 Command ===> _____  Scroll ===> PAGE
                                                    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          _____
 __  AZZCBLA1           YES      YES      YES          _____
 __  AZZCBLA2           YES      YES      YES          _____
 __  AZZCBL01           YES      YES                   _____
 __  AZZCBL02           YES      ------                _____
 X_  AZZCBY01           ------   YES                   _____  EXCLUDED
 X_  AZZEOFCD           ------   ------                _____  EXCLUDED
 X_  AZZSWTCH           ------   ------                _____  EXCLUDED
 ***************************** Bottom of data *****************************
```

Command line

There are several commands that can be entered on the command line. These commands will impact the list of members the user has driven.
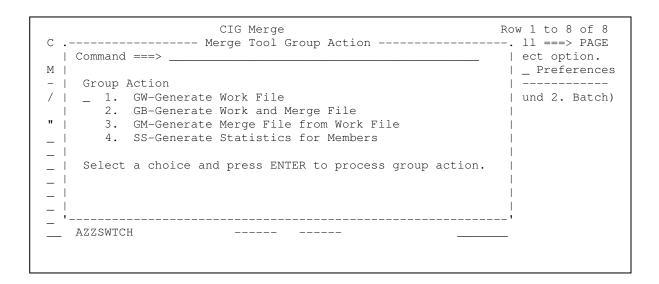
Command	Function
RES	Clear the "Action" column
X ALL	Exclude all members
CLEAR	Remove the member list

Group Actions

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

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

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

```
                      CIG Merge                          Row 1 to 8 of 8
C .----------------- Merge Tool Group Action -----------------. ll ===> PAGE
  | Command ===> _____  | ect option.
M |                                                           | _ Preferences
- | Group Action                                              | ------------
/ | _  1.  GW-Generate Work File                              | und 2. Batch)
  |    2.  GB-Generate Work and Merge File                    |
" |    3.  GM-Generate Merge File from Work File              |
_ |    4.  SS-Generate Statistics for Members                 |
_ |                                                           |
_ | Select a choice and press ENTER to process group action. |
_ |                                                           |
_ |                                                           |
_ |                                                           |
_ '-----------------------------------------------------------'
__   AZZSWTCH              ------    ------            _____
```

Action initiation

The Exec Mode field value controls the user that the user can initiate.

When the user specified "1", the Merge Tool behavior is based on the action supplied. In the example below a Generate Work file was requested. A progress indicator pop-up panel is displayed. The Function field is continually updated during the foreground processing.

```
                         CIG Merge Tool                      Row 1 to 8 of 8
C .--------------- Progress Indicator ---------------. _____    Scroll ===> PAGE
  |                                                   |  "/" to select option.
M |    Program : AZZCBLA2                             | ource Type _ Preferences
- |    Request : Build work file                      | -- --------------------
_ |    Function: Writing statistics and work file     |  1. Foreground 2. Batch)
  |                                                   | Source
" '---------------------------------------------------' Type
__   AZZASM01            YES       YES                    _____
__   AZZCBLA1            YES       YES                    _____
GW   AZZCBLA2            YES       YES                    _____
__   AZZCBL01            YES       YES                    _____
__   AZZCBL02            YES       ------                 _____
__   AZZCBY01            ------    YES                    _____
__   AZZEOFCD            ------    ------                 _____
__   AZZSWTCH            ------    ------                 _____
***************************** Bottom of data *****************************
```

Once the action is completed successfully, the note 'PROCESSED' is placed in the Action column and across from the member. Note also a YES appears in the Have Work column.

```
                         CIG Merge Tool                      Row 1 to 8 of 8
Command ===> _____    Scroll ===> PAGE
                                                 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                    _____
__   AZZCBLA1             YES     YES                    _____
GW   AZZCBLA2  PROCESSED  YES     YES      YES           _____
__   AZZCBL01             YES     YES                    _____
__   AZZCBL02             YES     ------                 _____
__   AZZCBY01             ------  YES                    _____
__   AZZEOFCD             ------  ------                 _____
__   AZZSWTCH             ------  ------                 _____
***************************** Bottom of data *****************************
```

When the user specifies "2", an Edit and Submit JCL panel is displayed. Note that the member being merged is displayed under the option line.

```
                            Merge Tool Edit and Submit JCL
   Option ===>

   PROGRAM: AZZCBL01

   Select one of the following options and press enter:

                          1    Edit JCL
                          2    Submit JCL

  Job cards:
  ======> //JOBID  JOB (YOURACCT),'NAME',
  ======> //     CLASS=?,REGION=2M,
  ======> //     MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=YOURID

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

                        END =  Exit   PF1 =  Help
```

From the Edit and Submit panel the user may edit or submit the generated JCL. Additionally, the user has the opportunity to add additional JCL to the Merge Tool built job.

When the user selects Submit JCL (2) the generated JCL is submitted and the user is returned to the Merge Tool Main panel. A note "SUBMITTED" is displayed in the Action column across from the member.

```
                   CIG Merge Tool                        Row 1 to 8 of 8

Command ===> _____  Scroll ===> PAGE
                                           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                    _____
 __  AZZCBLA1          YES     YES                    _____
 __  AZZCBLA2          YES     YES     YES            _____
 GW  AZZCBL01 SUBMITTED YES    YES                    _____
 __  AZZCBL02          YES     ------                 _____
 __  AZZCBY01          ------  YES                    _____
 __  AZZEOFCD          ------  ------                 _____
 __  AZZSWTCH          ------  ------                 _____
 ***************************** Bottom of data ******************************
```

When the user selects Edit JCL (1) the generated JCL is displayed in an edit panel. The user can modify the JCL and submit it from this panel.

```
 File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
 -------------------------------------------------------------------------------
 EDIT       P390P.SPFTEMP1.CNTL                              Columns 00001 00072
 Command ===>                                                  Scroll ===> PAGE
 ****** **************************** Top of Data *****************************
 000001 //P390PZZ  JOB (ACCT#),'CIG-INC',CLASS=A,REGION=4096K,
 000002 //      MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=P390P
 000003 //*************************************************
 000004 //*----------------------------------------------------------------*
 000005 //*  CIG MERGE TOOL FOR OS/390 AND Z/OS                            *
 000006 //*----------------------------------------------------------------*
 000007 //STEP1  EXEC PGM=IKJEFT01
 000008 //STEPLIB  DD  DSN=AZZ.V1R0.SAZZLOAD,DISP=SHR
 000009 //SYSPROC  DD  DSN=AZZ.V1R0.SAZZCLIB,DISP=SHR
```

To complete the edit session the user will PF3. The JCL has been updated and cannot be submitted using Submit JCL (2).

When the user exits the edit/submit panel, they are returned to the Merge Tool main panel and a "SUBMITTED" note is displayed in the Action column and across from the member.

```
                    CIG Merge Tool                          Row 1 to 8 of 8
 Command ===> _____    Scroll ===> PAGE
                                                    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                   _____
 __  AZZCBLA1          YES      YES                   _____
 __  AZZCBLA2          YES      YES      YES          _____
 GW  AZZCBL01 SUBMITTED YES     YES                   _____
 __  AZZCBL02          YES      ------                _____
 __  AZZCBY01          ------   YES                   _____
 __  AZZEOFCD          ------   ------                _____
 __  AZZSWTCH          ------   ------                _____
 ****************************** Bottom of data ******************************
```

27

When the action is SS the result will first be a foreground display of the member statistics.

```
File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
-------------------------------------------------------------------------------
VIEW       AZZ.V1R0.SAZZSAMS                            Columns 00001 00072
Command ===>                                             Scroll ===> PAGE
****** **************************** Top of Data ****************************
000001 16:05:59 ------ WORK FILE STATISTICS ------
000002 16:05:59 File1 name............ AZZ.V1R0.SAZZSAM1(AZZCBLA1)
000003 16:05:59 File2 name............ AZZ.V1R0.SAZZSAM2(AZZCBLA1)
000004 16:05:59 Baseline name........ AZZ.V1R0.SAZZSAMB(AZZCBLA1)
000005 16:05:59 # of lines in base.... 96
000006 16:05:59 # of lines in file1... 117
000007 16:05:59 # of lines in file2... 101
000008 16:05:59 # of unchanged lines.. 87
000009 16:05:59 # of inserts both..... 0
000010 16:05:59 # of inserts file1.... 22
000011 16:05:59 # of inserts file2.... 9
000012 16:05:59 # of deletes both..... 1
000013 16:05:59 # of deletes file1.... 5
000014 16:05:59 # of deletes file2.... 3
000015 16:05:59 # of conficts both.... 0
000016 16:05:59 # of conflicts file1.. 5
000017 16:05:59 # of conflicts file2.. 1
000018 16:05:59
****** **************************** Bottom of Data ****************************
```

When the user exits the Statistics display the 'PROCESSED' note is displayed in the Action column and across from the members.

When the user specifies Exec Mode 1 and an action of SS but has not defined the Statistics Files they will receive the "NEED DSN" message in the action column and across from the member.

```
                    CIG Merge Tool                        Row 1 to 8 of 8
 Command ===> _____  Scroll ===> PAGE
                                              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
 B1  AZZASM01  VIEWED    YES     YES                    _____
 __  AZZCBLA1            YES     YES                    _____
 EW  AZZCBLA2  EDITED    YES     YES      YES           _____
 SS  AZZCBL01  NEED DSN  YES     YES                    _____
 __  AZZCBL02            YES     ------                 _____
 E2  AZZCBY01  NOT FOUND ------  YES                    _____
 __  AZZEOFCD            ------  ------                 _____
 __  AZZSWTCH            ------  ------                 _____
 **************************** Bottom of data ****************************
```

If the user specifies an invalid combination for the Edit and Browse actions the appropriate message will display in the action column and across from the member.

An example of an invalid action would be specifying the E2 action for AZZCBY01 in the panel above. The "------" in the File2-Member column indicates no member named AZZCBY01 resides in File2. In this case E2 an invalid action for the member and the "NOT FOUND" message is displayed in the Action column.

Viewing statistics and log messages

The Merge Tool ISPF Workbench gives users the ability to review the statistics files as well as log messages file from the main panel. Enter "/" in the View Reports field.

```
                            CIG Merge Tool                        Row 1 to 8 of 8
 Command ===> _____   Scroll ===> PAGE
                                                  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                   _____
 __  AZZCBLA1             YES      YES                   _____
 __  AZZCBLA2  PROCESSED YES      YES        YES         _____
 __  AZZCBL01             YES      YES                   _____
 . . .
 ***************************** Bottom of data *****************************
```

Merge Tool will display a View Reports panel. The user selects one of the three choices by placing the corresponding number in the choice field and pressing enter.

```
                            CIG Merge Tool                        Row 1 to 8 of 8
 C .--------------------------- VIEW REPORTS ---------------------------.  PAGE
   | Command ===> _____ | ion.
 M |                                                                 | rences
 - |                                                                 | -----
 _ |  1  1.   Member Statistics                                      | Batch)
   |     2.   Summary Statistics                                     |
 " |     3.   Log Messages                                          |
 _ |                                                                 |
 _ |   Select a choice and press ENTER to view.                     |
 _ |                                                                 |
 _ |                                                                 |
 _ |                                                                 |
 _ |                                                                 |
 _ '-----------------------------------------------------------------'
 ***************************** Bottom of data *****************************
```

The types of reports are described in the following table.

#	File	Description
1	Member Statistics	The Member Statistics documents the number of modifications for each source module processed. Conflicting modifications are of particular significance. If there are conflicting changes, you will probably need to manually intervene making sure the merged source module will function as intended.
2	Summary Statistics	The Summary Statistics documents the totals related to number of members processed.
3	Log	The Log displays the parameters and statements in effect during the merge process. If processing does not complete successfully, the reason is reported here.

Please reference section Merge Tool Reports and Log for more detailed information about the Merge Tool reports.

When the user specifies 1, member statistics, the result will be a panel displaying the
member statistics.

```
 File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
 ------------------------------------------------------------------------------
 VIEW       AZZ.V1R0.SAZZSAMS                              Columns 00001 00072
 Command ===>                                              Scroll ===> PAGE
 ****** ***************************** Top of Data *****************************
 000001 09:48:39 ------ WORK FILE STATISTICS ------
 000002 09:48:39 File1 name............ AZZ.V1R0.SAZZSAM1(AZZCBLA2)
 000003 09:48:39 File2 name............ AZZ.V1R0.SAZZSAM2(AZZCBLA2)
 000004 09:48:39 Baseline name........ AZZ.V1R0.SAZZSAMB(AZZCBLA2)
 000005 09:48:39 # of lines in base.... 97
 000006 09:48:39 # of lines in file1... 92
 000007 09:48:39 # of lines in file2... 97
 000008 09:48:39 # of unchanged lines.. 88
 000009 09:48:39 # of inserts both..... 0
 000010 09:48:39 # of inserts file1.... 2
 000011 09:48:39 # of inserts file2.... 0
 000012 09:48:39 # of deletes both..... 0
 000013 09:48:39 # of deletes file1.... 7
 000014 09:48:39 # of deletes file2.... 0
 000015 09:48:39 # of conflcts both.... 0
 000016 09:48:39 # of conflicts file1.. 0
 000017 09:48:39 # of conflicts file2.. 2
 000018 09:48:39 --------------------------------
 ****** *************************** Bottom of Data ***************************
```

When the user specifies 2, summary statistics, the result will be a panel displaying
the summary statistics.

```
 File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
  ------------------------------------------------------------------------------
  VIEW       AZZ.V1R0.SAZZSAMZ                             Columns 00001 00072
  Command ===>                                             Scroll ===> CSR
  ****** ***************************** Top of Data ****************************
  000001 10:44:20 ------ SUMMARY STATISTICS ------
  000002 10:44:20 # of files existing in F1, F2, and Baseline.. 1
  000003 10:44:20 # of files existing in F1 and F2 only........ 0
  000004 10:44:20 # of files existing in F1 only............... 0
  000005 10:44:20 # of files existing in F1 and Baseline only.. 0
  000006 10:44:20 # of files existing in F2 and Baseline only.. 0
  000007 10:44:20 # of files existing in F2 only............... 0
  000008 10:44:20 # of files existing in Baseline only......... 0
  000009 10:44:20 # of files not found in any library.......... 0
  000010 10:44:20 --------------------------------
  ****** *************************** Bottom of Data ***************************
```

When the user specifies 3, log messages, the result will be a panel displaying the log
messages.

```
 File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
  ------------------------------------------------------------------------------
  VIEW       AZZ.V1R0.SAZZSAML                             Columns 00001 00072
  Command ===>                                             Scroll ===> CSR
  ****** ***************************** Top of Data ****************************
 000001 10:44:04 -------------------------------------------------
 000002 10:44:04 – CIG MERGE TOOL FOR OS/390 AND Z/OS
 000003 10:44:04 – Log file – 18 Feb 2002
 000004 10:44:04 -------------------------------------------------
 000005 10:44:18 Work file written: AZZ.V1R0.SAZZSAMW(AZZCBLA1)
 000006 10:44:18 Lines written....: 138
 000007 10:44:20 ------ SUMMARY STATISTICS ------
 000008 10:44:20 # of files existing in F1, F2, and Baseline.. 1
 000009 10:44:20 # of files existing in F1 and F2 only........ 0
 000010 10:44:20 # of files existing in F1 only............... 0
 000011 10:44:20 # of files existing in F1 and Baseline only.. 0
 000012 10:44:20 # of files existing in F2 and Baseline only.. 0
 000013 10:44:20 # of files existing in F2 only............... 0
 000014 10:44:20 # of files existing in Baseline only......... 0
 000015 10:44:20 # of files not found in any library.......... 0
 000016 10:44:20 --------------------------------
 ****** *************************** Bottom of Data ***************************
```

CHAPTER 3: THE CIG MERGE TOOL STEPS

Identify the merge components

The libraries where source versions are held

In order to properly merge the original source with the two independently modified versions of source or obtain statistical data to assist in the project management the user must first identify the merge components.

File	Description
Base File	The original or base source. This would be PDS with the original version of the source used for both sets of modifications.
File 1	The first of up to two PDS' containing the modified source. In the case of an emergency fix it might be the location of the fixed production source.
File 2	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.
Work File	A PDS that will hold the intermediate merged source. It must be seven bytes larger than the source being analyzed or merged.
Merge File	A PDS that will hold the result of the final merge action.

The member names

The user must determine if the PDS member names are the same between the three files. If the Base, File1 and File2 are the same dataset the will most likely be a member name mask, such as programmer initials, that prefixes the actual member name.

This information is important for the next step: Setting Merge Tool session Preferences as well as the Set Library panel.

Constructing the Work File member

Once the user has identified the files and member names the next step is to construct the Work File members.

Merging the changes into a Work File

To construct the work file via the Merge Tool ISPF interface the user should review and update the Set Libraries Panel to reflect the Base, File1, File2, Work and Merge files to reflect the PDS dataset names identified in the previous step.

Once the libraries are updated the member list can be driven. Using the Line Command or Group Action the user specifies GW (Generate Work File). Based on the Exec Mode field either a batch job is submitted (option 2) or the action is performed in foreground (option 1).

Note: To create a merged file in batch the user must specify both the work file and merged file. The merged file ddname is CIGMERGE.

Build Work Outputs

Upon successful completion of the Generate Work action the resulting member is placed in the Work File. If the member already exists it is replaced with the new Work File member. This output is an intermediate merge file. It contains the base (i.e., original) source with both sets of changes incorporated into the base. The member name will be the same as the File1 member name.

Statistical reports are also produced as output of the Build Work process. These reports identify number of members matched as well as reporting on number of lines inserted, deleted and conflicting changes.

Editing the Work File member

Upon successful creation of a Work File member the user now 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 it is determined that there is a conflict in overlapping changes the user 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.

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 four 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.
Column four and five indicate the origination of the modification (##): "1" indicating version one, "2" indicating version two, and "12" indicating that both versions contained the same modification.

A simple example of conflicting source changes follows.

```
File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
-------------------------------------------------------------------------------
EDIT      AZZ.V1R0.SAZZSAMW(AZZCBLA1) – 01.00          Columns 00001 00072
Command ===>                                           Scroll ===> CSR
****** **************************** Top of Data ****************************
000001 .---- **** CIG MERGE TOOL FOR OS/390 AND Z/OS       WORKFILE ****
000002 .---- BASELINE: AZZ.V1R0.SAZZSAMB(AZZCBLA1)
000003 .---- FILE 1  : AZZ.V1R0.SAZZSAM1(AZZCBLA1)
000004 .---- FILE 2  : AZZ.V1R0.SAZZSAM2(AZZCBLA1)
000005 .---- ....+....1....+....2....+....3....+....4....+....5....+....6....+.
000006              IDENTIFICATION DIVISION.
000007              PROGRAM-ID.   AZZCBLA1.
000008 .?+2         AUTHOR.        ARTHUR MURRAY.
000009 . +1         AUTHOR.        ARTHUR MILLER.
000010 . -12        AUTHOR.        ARTHUR AUTHOR.
000011              *******************************************************
000012              *
000013              * THIS IS A SAMPLE PROGRAM USED FOR THE MERGEIT TOOL.
000014              *
000015              *******************************************************
000016              ENVIRONMENT DIVISION.
000017              CONFIGURATION SECTION.
000018              INPUT-OUTPUT SECTION.
000019              FILE-CONTROL.
```

Creating the Merge File

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 will be one member with the combined changes, ready to add to the source management tool.

Merging the Work File

Using the Line Command or Group Action the user specifies GM (i.e., Generate Merge File). Based on the Exec Mode field either a batch job is submitted (option 2) or the action is performed in foreground (option 1).

Note: In batch, to create a merge file in batch directly from a work file specify the ddnames CIGWORK and CIGMERGE only, CIGBASE, CIGDD01, and CIGDD02 will be omitted.

If CIGBASE, CIGDD01, and CIGDD02 are specified then the work file specified in CIGWORK will be replaced prior to the creation of the final merged file.

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 Work File member. The member name will be the same as the Merge File member name.

The output is the combined merge file. It contains the base (i.e., original) source with both sets of changes incorporated into the base, based on the merge directives in columns one through 4 of the Work File.

When addressing conflicts in the Work File the user may determine that the change, although in the same section of code do not overlap. It may be determined that all the changes should be merged, even lines in conflict.

The example below shows what the merge process will do if conflicts are not resolved. Note that both author statements were included in the Merge File source.

```
File   Edit   Confirm   Menu   Utilities   Compilers   Test   Help
--------------------------------------------------------------------------
EDIT       AZZ.V1R0.SAZZSAMM(AZZCBLA1) – 01.00          Columns 00001 00072
Command ===>                                            Scroll ===> CSR
****** **************************** Top of Data ****************************
000001        IDENTIFICATION DIVISION.
000002        PROGRAM-ID.    AZZCBLA1.
000003        AUTHOR.        ARTHUR MURRAY.
000004        AUTHOR.        ARTHUR MILLER.
000005        *****************************************************************
000006        *
000007        * THIS IS A SAMPLE PROGRAM USED FOR THE MERGEIT TOOL.
000008        *
000009        *****************************************************************
000010         ENVIRONMENT DIVISION.
000011         CONFIGURATION SECTION.
000012         INPUT-OUTPUT SECTION.
000013         FILE-CONTROL.
000014
000015            SELECT REC-IN ASSIGN TO OUTFILE
000016            FILE STATUS IS SF-STAT.
000017
000018            SELECT NAME-IN ASSIGN TO NAMEFILE
000019            FILE STATUS IS SF-STAT.
```

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

Test the outputs.

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

34

CHAPTER 4: THE CIG MERGE TOOL BATCH UTILITY

The JCL

The Merge Tool can be executed via a batch job. Three sample JCL members can be found in the AZZ.SAZZJCL library created during the installation procedure.

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.

```
//*JOBCARD INFO
//*------------------------------------------------------------------*
//*  AZZMRKUP JCL                                                     *
//*------------------------------------------------------------------*
//STEP1  EXEC PGM=IKJEFT01
//STEPLIB  DD  DSN=AZZ.SAZZLOAD,DISP=SHR
//SYSTSIN  DD  *
  EXEC 'AZZ.SAZZCLIB(MERGE)'
//SYSTSPRT DD  SYSOUT=*
//*
//*  TO BUILD A WORK FILE,
//*  ALLOCATE CIGBASE, CIGDD01, CIGDD02, AND CIGWORK TO
//*
//*  NOTE: CIGWORK MUST HAVE A LRECL 8 LARGER THAN CIGDD01
//*
//CIGBASE  DD  DISP=SHR,DSN=QUAL1.QUAL2.PDS
//CIGDD01  DD  DISP=SHR,DSN=QUAL1.QUAL2.1.PDS
//CIGDD02  DD  DISP=SHR,DSN=QUAL1.QUAL2.2.PDS
//CIGWORK  DD  DISP=SHR,DSN=QUAL1.QUAL2.WORK
//*
//*  IF DEBUG/TRACING IS DESIRED THEN THE REQUEST MUST
//*  BE THE FIRST LINE IN CIGIN FILE.
//*
//*  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
       COL 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=*
```

35

AZZMERGE – Build the Final Merge File

The following JCL is a sample of the AZZMERGE 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.

```
//*JOBCARD INFORMATION
//*-----------------------------------------------------------------*
//*   AZZMERGE JCL                                                   *
//*-----------------------------------------------------------------*
//STEP1   EXEC PGM=IKJEFT01
//STEPLIB  DD  DSN=AZZ.SAZZLOAD,DISP=SHR
//SYSTSIN  DD  *
  EXEC 'AZZ.V0R1.SAZZCLIB(AZZMERGE)'
//SYSTSPRT DD  SYSOUT=*
//*
//*  TO BUILD A MERGE FILE FROM A WORK FILE,
//*  ALLOCATE CIGMERGE.
//*
//*  NOTE: CIGWORK MUST HAVE A LRECL 8 LARGER THAN CIGDD01
//*
//CIGWORK  DD  DISP=SHR,DSN=QUAL1.QUAL2.WORK
//CIGMERGE DD  DISP=SHR,DSN=QUAL1.QUAL2.MERGE
//*
//*  IF DEBUG/TRACING IS DESIRED THEN THE REQUEST MUST
//*  BE THE FIRST LINE IN CIGIN FILE.
//*
//*  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)
//*
//*  THE FOLLOWING LINE IS COMMENTED OUT FOR NOW / NO LOGIC TO SUPPORT
//*
//CIGIN    DD  *
SELECT FILE1 XXXXXXXX
       FILE2 XXXXXXXX
       BASE XXXXXXXX
       COL 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=*
```

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 will create a Work File and immediately interpret the Work File, creating a Merged File with both sets of changes incorporated, including conflicting changes.

```
//*JOBCARD INFO
//*------------------------------------------------------------------*
//*  AZZMRKUP JCL                                                     *
//*------------------------------------------------------------------*
//STEP1  EXEC PGM=IKJEFT01
//STEPLIB  DD  DSN=AZZ.SAZZLOAD,DISP=SHR
//SYSTSIN  DD  *
  EXEC 'AZZ.V0R1.SAZZCLIB(MERGE)'
//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 8 LARGER THAN CIGDD01
//*
//CIGBASE  DD  DISP=SHR,DSN=QUAL1.QUAL2.PDS
//CIGDD01  DD  DISP=SHR,DSN=QUAL1.QUAL2.1.PDS
//CIGDD02  DD  DISP=SHR,DSN=QUAL1.QUAL2.2.PDS
//CIGWORK  DD  DISP=SHR,DSN=QUAL1.QUAL2.WORK
//CIGMERGE DD  DISP=SHR,DSN=QUAL1.QUAL2.MERGE
//*
//*  IF DEBUG/TRACING IS DESIRED THEN THE REQUEST MUST
//*  BE THE FIRST LINE IN CIGIN FILE.
//*
//*  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)
//*
//*  THE FOLLOWING LINE IS COMMENTED OUT FOR NOW / NO LOGIC TO SUPPORT
//*
//CIGIN    DD  *
SELECT FILE1 XXXXXXXX
      FILE2 XXXXXXXX
      BASE XXXXXXXX
      COL 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=*
```

Utility DD statements

The DD statements used in the previous three examples are documented below.

DDNAME	DESCRIPTION
CIGBASE	The Base File or original source used for modifications. This DD is required for the GW and GB actions.
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 and GB actions.
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 and GB actions.
CIGWORK	The Work File library. This is where the 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.
CIGMERGE	The Merge File library. This is where the Merge Tool puts the output from the final merge step (i.e., Generate Merge). This DD is required for the GM and GB actions.
CIGSTATS	Number of modifications for each source module processed. Conflicting modification are of particular significance. This DD is required for statistical reporting and may be a dataset or SYSOUT.
CIGSUMM	Totals related to number of members processed. This DD is required for summary reporting and may be a dataset or SYSOUT.
CIGLOG	Parameters and statements in effect during the merge process. If processing fails, the reason is reported here. Traces are also written to this DD when requested and may be a dataset or SYSOUT.

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.

```
SELECT    FILE1 File1 member-name
          FILE2 File2 member-name
          BASE base member-name
          COLTYPE compare-type
```

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 the Merge Tool requests is defined in the following table.

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.	
COLTYPE	JCL DATA ASM COBOL PL1 XX:XX (s:e)	Type of source being merged. This will override the default merge compare columns (all source columns) to ones specific to the type. Example: COBOL will direct the Merge Tool to compare source columns 7 – 72. This parameter is optional. Default is all columns.

SYNTAX EXAMPLES:

Example1:
SELECT FILE1 *

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

Example2:
SELECT FILE1 AZZCBY* FILE2 AZZCBZ* BASE AZZCBL* COLUMN TYPE COBOL

Example2 requests to use the member name pattern to match the member names of FILE1 that begin with CBY and members in FILE2 that begin with CBZ with BASE members that begin with CBL. Additionally, the COLUMN TYPE was specified as COBOL so only columns 7 – 72 will be compared.

Example3:
SELECT FILE1 AZZCBL01

Example3 requests that the Merge Tool use the member name specified in the FILE1 parameter for the FILE2 and BASE files as object for the action.

CHAPTER 5: CIG MERGE TOOL REPORTS AND STATISTICS

Types of reports available

There are a total of three different types of reporting the Merge Tool performs. The types of reports and descriptions follow.

REPORT TYPE	DESCRIPTION
CIGSTATS	Number of modifications for each source module processed. Conflicting changes are of particular significance. This DD is required for Mark-up statistical reporting and may be a dataset or SYSOUT.
CIGSUMM	Totals related to number of members processed. This DD is required for Mark-up summary reporting and may be a dataset or SYSOUT.
CIGLOG	Parameters and statements in effect during the merge process. If processing fails, the reason is reported here. Traces are also written to this DD when requested and may be a dataset or SYSOUT.

Samples of the reports follow.

Sample Report output for single member

Summary Statistics report for single member Mark-up request

```
000001 09:48:39 ------ SUMMARY STATISTICS ------
000002 09:48:39 # of files existing in F1, F2, and Baseline.. 1
000003 09:48:39 # of files existing in F1 and F2 only........ 0
000004 09:48:39 # of files existing in F1 only............... 0
000005 09:48:39 # of files existing in F1 and Baseline only.. 0
000006 09:48:39 # of files existing in F2 and Baseline only.. 0
000007 09:48:39 # of files existing in F2 only............... 0
000008 09:48:39 # of files existing in Baseline only......... 0
000009 09:48:39 # of files not found in any library.......... 0
000010 09:48:39 -------------------------------
```

Member Statistics report for single member Mark-up request

```
09:48:39 ------ WORK FILE STATISTICS ------
09:48:39 File1 name............ AZZ.V1R0.SAZZSAM1(AZZCBLA2)
09:48:39 File2 name............ AZZ.V1R0.SAZZSAM2(AZZCBLA2)
09:48:39 Baseline name......... AZZ.V1R0.SAZZSAMB(AZZCBLA2)
09:48:39 # of lines in base.... 97
09:48:39 # of lines in file1... 92
09:48:39 # of lines in file2... 97
09:48:39 # of unchanged lines.. 88
09:48:39 # of inserts both..... 0
09:48:39 # of inserts file1.... 2
09:48:39 # of inserts file2.... 0
09:48:39 # of deletes both..... 0
09:48:39 # of deletes file1.... 7
09:48:39 # of deletes file2.... 0
09:48:39 # of conficts both.... 0
09:48:39 # of conflicts file1.. 0
09:48:39 # of conflicts file2.. 2
09:48:39 -------------------------------
```

Log activity report for single member Mark-up request

```
09:48:26 -------------------------------------------
09:48:26 - CIG MERGE TOOL FOR OS/390 AND Z/OS
09:48:26 - Log file - 18 Feb 2002
09:48:26 -------------------------------------------
09:48:38 Work file written: AZZ.V1R0.SAZZSAMW(AZZCBLA2)
09:48:38 Lines written....: 104
09:48:39 ------ SUMMARY STATISTICS ------
09:48:39 # of files existing in F1, F2, and Baseline.. 1
09:48:39 # of files existing in F1 and F2 only........ 0
09:48:39 # of files existing in F1 only............... 0
09:48:39 # of files existing in F1 and Baseline only.. 0
09:48:39 # of files existing in F2 and Baseline only.. 0
09:48:39 # of files existing in F2 only............... 0
09:48:39 # of files existing in Baseline only......... 0
09:48:39 # of files not found in any library.......... 0
09:48:39 --------------------------------
```

Sample Report output for full library request

Summary Statistics for full library Mark-up request

```
------ SUMMARY STATISTICS ------
# of files existing in F1, F2, and Baseline.. 4
# of files existing in F1 and F2 only........ 1
# of files existing in F1 only............... 2
# of files existing in F1 and Baseline only.. 1
# of files existing in F2 and Baseline only.. 1
# of files existing in F2 only............... 0
# of files existing in Baseline only......... 1
# of files not found in any library.......... 0
```

Log activity report for full library Mark-up request

```
12:43:51 -------------------------------------------------
12:43:51 - CIG MERGE TOOL FOR OS/390 AND Z/OS
12:43:51 - Log file - 13 Feb 2002
12:43:51 -------------------------------------------------
12:44:03 Work file written: AZZ.V0R1.SAZZSAMW(AZZCBLA1)
12:44:03 Lines written....: 106
12:44:03 Lines written....: 110
12:44:03 ------ SUMMARY STATISTICS ------
12:44:03 # of files existing in F1, F2, and Baseline.. 4
12:44:03 # of files existing in F1 and F2 only........ 1
12:44:03 # of files existing in F1 only............... 2
12:44:03 # of files existing in F1 and Baseline only.. 1
12:44:03 # of files existing in F2 and Baseline only.. 1
12:44:03 # of files existing in F2 only............... 0
12:44:03 # of files existing in Baseline only......... 1
12:44:03 # of files not found in any library.......... 0
```

Member Statistics for full library Mark-up request

```
15:13:06 ------ WORK FILE STATISTICS ------
15:13:06 File1 name............ AZZ.V1R0.SAZZSAM1(AZZCBLA1)
15:13:06 File2 name............ AZZ.V1R0.SAZZSAM2(AZZCBLA1)
15:13:06 Baseline name......... AZZ.V1R0.SAZZSAMB(AZZCBLA1)
15:13:06 # of lines in base.... 96
15:13:06 # of lines in file1... 117
15:13:06 # of lines in file2... 101
15:13:06 # of unchanged lines.. 85
15:13:06 # of inserts both..... 0
15:13:06 # of inserts file1.... 29
15:13:06 # of inserts file2.... 9
15:13:06 # of deletes both..... 2
15:13:06 # of deletes file1.... 4
15:13:06 # of deletes file2.... 3
15:13:06 # of conficts both.... 0
15:13:06 # of conflicts file1.. 0
15:13:06 # of conflicts file2.. 2
15:13:06 ------------------------------
15:13:07 ------ WORK FILE STATISTICS ------
15:13:07 File1 name............ AZZ.V1R0.SAZZSAM1(AZZCBLA2)
15:13:07 File2 name............ AZZ.V1R0.SAZZSAM2(AZZCBLA2)
15:13:07 Baseline name......... AZZ.V1R0.SAZZSAMB(AZZCBLA2)
15:13:07 # of lines in base.... 97
15:13:07 # of lines in file1... 92
15:13:07 # of lines in file2... 97
15:13:07 # of unchanged lines.. 88
15:13:07 # of inserts both..... 0
15:13:07 # of inserts file1.... 2
15:13:07 # of inserts file2.... 0
15:13:07 # of deletes both..... 0
15:13:07 # of deletes file1.... 7
15:13:07 # of deletes file2.... 0
15:13:07 # of conficts both.... 0
15:13:07 # of conflicts file1.. 0
15:13:07 # of conflicts file2.. 2
15:13:07 ------------------------------
15:13:08 ------ WORK FILE STATISTICS ------
15:13:08 File1 name............ AZZ.V1R0.SAZZSAM1(AZZCBL01)
15:13:08 File2 name............ AZZ.V1R0.SAZZSAM2(AZZCBL01)
15:13:08 Baseline name......... AZZ.V1R0.SAZZSAMB(AZZCBL01)
report continues…
```

This report will continue with an entry for each member that was used as object to the action.

42

APPENDIX A – SAMPLE CIG MERGE TOOL SCENARIOS

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

If the site is using the SCLM Suite the Base Libraries can be defined using the SCLM life cycle concatenation (perhaps TEST and RELEASE libraries) and File1 & File2 may be the development libraries, housing the independently changed source members. In Addition the Merge File may also be identified as one of the SCLM libraries. Once the final Merge action is completed the newly merged source can be migrated (registering the member to SCLM). Finally, a Build can be performed creating new outputs to test for all functionality.

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 will reverse 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 the CIG Merge tool.
Assumption: The CIG 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 the CIG 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 step1.

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

Note1: This can also be accomplished by typing GW across from the member on the main Merge Tool panel.

Note2: If the changes are of a simple nature, the programmer can use the GM action to generate the Work file and the Merge file, in one step, skipping the next step.

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 "1" for Generate Work file.
   a. Source lines can be inserted into or deleted from the work file.
   b. Merge directives can be changed (i.e., cc 1 – 7 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 l list and "3" for Generate Merge file from Work file.

8. Once the source is consolidated, the programmer will access their source management tool 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 may 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 may 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 will cause source regression and production failures to return.

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

Assumption: The CIG 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 will be identified as File1 and what source will be identified as File2. Identify the location of the:
   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 the CIG 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 step1.

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

   Note1: This can also be accomplished by typing GW across from the member on the main Merge Tool panel.
   Note2: If the changes are of a simple nature, the programmers can use the GM action to generate the Work file and the Merge file, in one step, skipping the next step.

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 (i.e., 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 "1" for Generate Work file.
   a. Source lines can be inserted into or deleted from the work file.
   b. Merge directives can be changed (i.e., cc 1 – 7 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 l list and "3" for Generate Merge file from Work file.

8. Once the source is consolidated, the programmers will access the source management tool (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 may 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 may 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 will reverse 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 the 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 the CIG Merge Tool.

Assumption: the CIG 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 the CIG 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 step1.

3. Omitting member name, build member list by pressing Enter.

4. Select reports and run member statistics, and summary statistics against the appropriate files.

   Note: For foreground this will require Library data set updates between each submit if the members are spread across several data set types (i.e. COBOL and Copybooks in one set of libraries, JCL and PROC source in another set of libraries, etc.). In Batch JCL this is a change to the CIGDD01, CIGDD02 and CIGWORK DD statements of the MERGEIT 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 (i.e., inserts, deletes, and conflicting changes).

6. Based on analysis Project Management assigns merge tasks to programmers.

Programmer actions follow:

7. 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.

8. In TSO invoke the CIG 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 step1.

9. Omitting member name, build member list by pressing Enter.

10. Select GROUP ACTION with "/" for action list and "1" to generate the Work file

    Note1: This can also be accomplished by typing GW in the Group Action field on the main Merge Tool panel.

    Note2: If the changes are of a simple nature, the programmers can use the GM group action to generate the Work file and the Merge file, in one step, skipping the next step.

11. 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.

12. 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 "1" for Generate Work file.
   a. Source lines can be inserted into or deleted from the work file.
   b. Merge directives can be changed (i.e., cc 1 – 7 of the work file).
   c. All conflicting lines can remain and be included in Merge output; this may be done to test to see if both sets of changes can coexist.

13. 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 l list and "3" for Generate Merge file from Work file.

14. Once the source is consolidated, the programmers will access the source management tool 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 may be edited and merged again, creating new output to test (steps 12, 13 and 14). These steps are repeated until successful testing is accomplished.

15. Once the programmers have successfully tested the source for all conditions the code may 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, it should not be necessary to entirely re-code the customization.

Resolution: Using the CIG 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 the CIG Merge tool.

Assumption: The CIG 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 should 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 the CIG Merge Tool allows the user to append to lists of members and override member type (e.g., 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:

3. 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).

4. Invoke the CIG 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 step1.

50

5. Omitting member name, build member list by pressing Enter.

6. Select reports and run member statistics, and summary statistics against the appropriate files.

   Note: For foreground this will require Library data set updates between each submit if the members are spread across several data set types (i.e. COBOL and Copybooks in one set of libraries, JCL and PROC source in another set of libraries, etc.). In Batch JCL this is a change to the CIGDD01, CIGDD02 and CIGWORK DD statements of the MERGEIT JCL.

7. 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 (i.e., inserts, deletes, and conflicting changes).

8. Based on analysis Project Management assigns merge tasks to programmers.

Programmer actions follow:

9. 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 (i.e., the upgrade) the vendor shipped requiring the retrofit of the customization.

10. Invoke the CIG 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 step1.

11. Omitting member name, build member list by pressing Enter.

12. Select GROUP ACTION with "/" for action list and "1" to generate the Work file.

    Note1: This can also be accomplished by typing GW in the Group Action field on the main Merge Tool panel.

    Note2: If the changes are of a simple nature, the programmers can use the GM group action to generate the Work file and the Merge file, in one step, skipping the next step.

13. Programmer reviews Work File and statistics and determines which members require retrofitting and the complexity of the effort to consolidate the two versions.

14. 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 "1" for Generate Work file.
    a. Source lines can be inserted into or deleted from the work file.
    b. Merge directives can be changed (i.e., cc 1 – 8 of the work file).
    c. All conflicting lines can remain and be included in Merge output; this may be done to test to see if both sets of changes can coexist.

Note: Changes indicated as made by File2 will be the changes for the new release and in most cases will remain. These changed should be ignored unless they overlap the customization (File1).

15. 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 l list and "3" for Generate Merge File from Work File.

16. Once the source is merged, the programmers will access the source management tool 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 may be edited and merged again, creating the new outputs to test (steps 14, 15 and 16). These steps are repeated until successful testing is accomplished.

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