CA Verify® Automated Regression Testing for CICS

User Guide

Version 9.0.00

CA Technologies Product References

This document references the following CA Technologies products:

CA Verify® Automated Regression Testing for CICS (CA Verify for CICS)

CA Verify® Automated Regression Testing for VTAM (CA Verify for VTAM)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 7: Edit Function 133

Chapter 8: Rules Function 151

Chapter 11: Batch Functions 247

Chapter 12: Commands for REXX 275

Chapter 1: Introduction

CA Verify Automated Regression Testing for CICS (CA Verify for CICS), an automated quality assurance testing tool, ensures that new or revised CICS application and system changes are thoroughly tested so they function correctly in production.

CA Verify for CICS *automates* testing by capturing a sequence of input and output screens from live CICS transactions. The captured screens constitute a *test stream*. After an application or system change, you can re-execute the test stream. CA Verify for CICS compares the current output screens with the original output screens and displays all differences (*mismatches*) online. You can also use CA Verify for CICS to run test streams in batch and print the differences.

CA Verify for CICS also permits *rules* and *rulesets* to be defined and applied to captured test streams. The rules notify CA Verify for CICS in advance of expected test stream differences and how to handle them. The rules feature makes it easy to run captured test streams against new releases of an application, by eliminating the need to stop for expected changes.

This section contains the following topics:

Benefits of CA Verify for CICS

Here are some of the benefits when you use CA Verify for CICS for testing:

- You will not have to re-key data during test sessions because CA Verify for CICS saves the original input screens with the data already entered.
- You will not have to compare test results manually. Manual *before* and *after* comparisons, always tedious and error-prone, are eliminated.
- Once you have captured a test stream, you can modify it directly or establish rules that model and handle expected changes, as your testing needs change.
- You can simulate production conditions — for example, you can use real, rather than contrived, test data
  - Execute similar or identical transactions which try to perform the same task at the same time
  - Simulate high-volume activity *without* tying up system resources or affecting response time because CA Verify for CICS uses *virtual* terminals

Types of Testing

CA Verify for CICS is the ideal tool for all of the following types of testing:

Unit:

Test a program change and see how it affects the rest of the program

Integration:

Test whether a modified program works properly in conjunction with other programs

Concurrency:

Test what happens when two or more transactions try to perform the same task at the same time; for example, update a record

Regression:

Test whether a system performs as usual after a change to one component of that system

Stress:

Test how your system behaves under heavy loads and determine how increased transaction volume affects response time

Migration:

Test whether existing applications perform as expected following hardware or system changes; for example, a new disk drive or a new release of CICS

See the "Testing" chapter (see page 29) for suggestions on how to perform different types of testing.

Security Considerations

Because CA Verify for CICS is a CICS application, it is subject to whatever CICS security system you are using. In addition, CA Verify for CICS performs its own internal checking based on operator ID to ensure that test streams are accessed only by authorized users.

CA Verify for CICS also has these safeguards:

- You can specify read, write, or print protection for test streams and rulesets.
- You can specify that CA Verify for CICS not display the contents of dark (non-display) screen fields.
- Specific transactions can be globally excluded from test streams. Moreover, when you create an individual test stream, you can specify which transactions to include or exclude.
- You can include/exclude transactions by user ID.

Additional Features

Other important features include:

- A user exit interface that lets you customize CA Verify for CICS. For example, you can call an external security system or modify screen data when a test stream is executed.
- The ability to process command lists which can be used to open files, verify CICS initialization, and perform CICS termination tasks.
- A powerful rules function to give advance notification to CA Verify for CICS of expected test stream differences and how to handle them. This feature is particularly useful in filtering out the anticipated changes of a new release from the unexpected changes.
- The ability to convert test streams to REXX and execute them under a CA Verify REXX Command Environment.
- An online demo session that helps you quickly learn how to use CA Verify for CICS.
- Menus and PF keys which follow ISPF standards, so new users find it easy to use.
- An automated signon and signoff capability which eliminates the need to capture required security transactions in every test stream.
- A TSO interface that lets you use the Browse, Edit, Maintain Rules, and Utility functions under TSO.

How to Get Started

If you are a new user, read the "Getting Started (see page 19)" and "Testing (see page 29)" chapters and then perform the demo session located in the "Installation Verification Procedure and Demo Session (see page 355)" chapter. Then you will have all the information you need to begin creating your own test streams.

To perform simple tests, you'll only need to see the following chapters:

- "Log Function (see page 37)" for instructions on creating a test stream.
- "Run Function (see page 63)" for instructions on executing a test stream.

Consult the "Browse Function (see page 111)" chapter to learn how to examine test streams online.

To begin using the rules function, consult the *Rules Primer*, and the "Rules Function (see page 151)" chapter in this guide.

As you become more expert in using CA Verify for CICS, you will want to see other chapters which explain more complex functions.

Chapter 2: Getting Started

This section contains the following topics:

Invoke CA Verify for CICS

Follow these steps:

1. Clear your screen and type the transaction code:

   XTCA

   CA Verify for CICS displays the Primary Options Menu.

```
VERSION x.x.x---------- CA VERIFY PRIMARY OPTIONS MENU  ----------10:29:28
ENTER COMMAND ===>                                          TERM: L9D6CA
                                                            OPER: NGN
   L  LOG A TEST STREAM

   B  BROWSE A TEST STREAM

   R  RUN A TEST STREAM

   E  EDIT A TEST STREAM

   M  MAINTAIN RULES

   I  INQUIRY/TERMINATION OF FUNCTIONS

   U  UTILITIES

   T  TUTORIAL

   X  EXIT
          *** ENTER "KEYS" COMMAND TO SEE PF KEY ASSIGNMENTS ***


COPYRIGHT © 2012 CA.    ALL RIGHTS RESERVED
```

   The previous menu lists the functions, which are described in the next section.

2. To select a function, type its letter in the Enter Command field and press Enter.

Bypass the Primary Options Menu

Follow these steps:

1. Specify an option when you invoke CA Verify for CICS.

   For example:

   XTCA T

   Accesses the Tutorial.

   You can also access a function without returning to the Primary Options Menu by typing =x on the command line of any menu except those displayed when a test stream is running. Replace x with the letter identifying the function.

   For example:

   =B

   Accesses the Browse function.

Function Overview

Log a Test Stream

The *Log* function is the process by which CA Verify for CICS captures input screens as they are received from the terminal and output screens as they are sent to the terminal. The captured records constitute a *test stream*.

Browse a Test Stream

The *Browse* function lets you examine a test stream you have logged or run. You can look at any of the screens and any information saved when the test stream was run.

Run a Test Stream

The *Run* function is the process by which CA Verify for CICS re-executes a test stream. CA Verify for CICS sends the captured input screens to the application, compares the output screens generated by the application with the original output screens, and flags any differences.

Edit a Test Stream

The *Edit* function lets you modify a test stream you have logged or run. For example, you can delete screens or change input data.

Maintain Rules

The Maintain Rules function lets you create, modify, and delete rulesets and rules that can be applied to one test stream, all test streams within an application, and/or all test streams in a system. Rulesets and rules tell CA Verify for CICS, in advance, what changes you expect to see when running a test stream in a modified application or environment. When using rulesets, CA Verify for CICS will not flag the differences accounted for by the rules, which greatly streamlines the testing of new releases with existing test streams.

In addition to controlling the changes made within a test stream, you can use the Rules function while logging to control which terminals, transactions, or users should be logged.

Inquiry and Termination of Active Functions

The *Inquiry* function lets you display the status of all active and suspended functions and *Terminate* Log functions.

Utilities

The *Utilities* let you manipulate test streams. Using the Utilities, you can:

- Update a test stream's directory information
- Append screens from one test stream to the end of another test stream
- Insert screens from one test stream into another test stream
- Merge two test streams
- Copy a test stream from one name to a new name, and optionally copy its ruleset at the same time
- Rename a test stream and its associated ruleset
- Delete a test stream and its associated ruleset

Tutorial

Use the *Tutorial* to learn how to use CA Verify.

Exit CA Verify for CICS

Use the *Exit* function to exit from CA Verify for CICS and return to CICS.

Types of Screens

While using CA Verify for CICS, you will encounter three types of screens:

Original Screen

Displays the record from the originally logged test stream with no changes applied.

Expected Screen

Displays the original screen with rules applied.

Current Screen

Shows the screen that results from the test stream being run.

In most cases, you can use PF2 (Rotate) to switch the display from Expected to Current to Original and then back to Expected.

Test Stream Contents

Each record in a test stream created by the Log function contains either an input or output screen. Records in test streams created (or updated) by the Run function may also contain additional information: an original screen or mismatch signoff comments.

Each test stream record consists of one or more parts, as illustrated following:

Input Record	Output Record
Current input screen	Current output screen
Original input screen (optional)	Original output screen (optional)
	Mismatch Data (optional)
	Applied Rules (optional)
	Expected Screen (optional)

Input Record

Current input screen

The current input screen always appears in a test stream. It is the input from the log or most recent run.

Original input screen

The original input screen appears in a test stream created by the Run function only if it differs from the current input screen and record history was specified.

Output Record

Applied Rules Screen

Shows the rules applied to the original screen to obtain the expected screen. This screen is similar to the Rules Applied During Run screen.

Current Output Screen

The current output screen always appears in the test stream. It is the output from the log or most recent run.

Expected Output Screen

The original output screen showing any rules that have been applied.

Original Output Screen

The original output screen appears in a test stream created by the Run function only if it differs from the current output screen and record history was specified.

Signoff Data

This information is specified by you during the Run function to explain the mismatch.

Mismatch signoff information is available only if this option was specified during the most recent run.

Record Numbers

A record number identifies every record in the test stream. This is a sequential number associated with each current input or output screen. For example, if a test stream consists of one input screen, followed by one output screen, followed by one input screen, followed by one output screen, all input screens will be odd numbered and all output screens will be even numbered.

Record Numbers

A record number identifies every record in the test stream. This is a sequential number associated with each current input or output screen. For example, if a test stream consists of one input screen, followed by one output screen, followed by one input screen, followed by one output screen, all input screens will be odd numbered and all output screens will be even numbered.

Types of Comparisons

When CA Verify for CICS runs a test stream, it compares the original output screens with the current screens. You can specify one of the following comparison types:

Screen

Compares corresponding rows of two screens to see if they're the same; comparison base includes all output screens between two input screens.

This type of comparison must be used when processing a test stream with a ruleset.

Logical

Compares the corresponding rows on the original and current screens to see if they're the same. Variation in bytes transmitted to the terminal is allowed as long as the end result is the same.

A logical comparison also allows for variable fields (that is, fields you would expect to be different, such as run date or time-of-day).

Note: A logical comparison is not supported for non-3270 terminals.

Physical

Compares 80-byte segments of the data streams.

Use this type of comparison for non-3270 terminal test streams and for test streams that use graphics.

During a logical, physical, or screen comparison, CA Verify for CICS halts the run if it detects any difference between the original and current screens. At this point you have many different options. For example, you can ignore the mismatch, define additional rules, or terminate the run.

Panels

Most panels have the same header format. A sample panel is shown following, followed by an explanation of the common areas.

```
1                          2                                        3
CCC.ORDERAPP.001     ---  BROWSE: DIRECTORY INFORMATION  ----INVALID COMMAND
ENTER COMMAND ===> XXX 4
    5
 DESCRIPTION: CAROL'S COOKIES COMPANY DEMO PROGRAM



            LOG:       RUN:       EDIT:       TEST STREAM:     IN: OUT:
 INVOKED BY: SAB                   SAB         TOTAL SCREENS:    4    4
 INVOKED ON: 05/22/1998            05/22/1998  AVERAGE BYTES:    8  442
 START TIME: 13:56:00              14:03:45
 DURATION:   00:02:08   00:00:01
 SYSTEM:     CICSC9AJ
 STATUS:     NORMAL
 TERMINAL:   NN01
 VSAM CI'S:  1

 AVERAGE THINK TIME:    00:00:31.817
 AVERAGE RESPONSE TIME: 00:00:00.248
 MAXIMUM SCREEN SIZE:   24 BY  80
 PROTECTION STATUS:

 PRESS ENTER TO CONTINUE OR PF3 TO END
```

1 Name area

Indicates the name of the test stream when a specific test stream is being processed; otherwise, it contains dashes

2 Title

Identifies the menu

3 Message area

Contains an error or informational message; otherwise, contains the time

4 Command area

Use this field to type a command, option, or parameter

5 Extended message area

Displays an extended, numbered message if you type the Help (PF1) command when an error message appears in the message area **3**. At other times, instructions on how to proceed will appear.

Commands and PF Keys

Use the following commands with all functions. Commands that apply to a specific function are discussed in the appropriate chapters. Type all commands except Redisplay in the command area.

Where applicable, the default PF key which can be used instead of the command is listed.

Caps On/Off

Translates lowercase characters to uppercase on menus with the following fields which support lowercase:

Edit function

Field and Hex format input

Browse and Edit functions

Find and Change command parameters

Rules function

All values and description fields

Log and Utility functions

Terminal names

End (PF3)

Ends processing of the current menu and returns to prior processing level.

Help (PF1)

If a brief error message is displayed, replaces that message with a more detailed, numbered message on the third line of the menu. If an error message is not displayed, displays the Tutorial panel for the current function.

Keys (PF12)

Displays a list of PF key assignments.

OPTS

Displays the options as currently set in the OPTIONS module.

Redisplay (PA2, CLEAR, PFn)

Refreshes the last menu displayed. Invoke this command by pressing the PA2 key, Clear, or the assigned PF key.

Resume

Continues processing from the point of suspension. This command is valid *only* from the Primary Options Menu. See Suspend.

Return (PF4)

Ends processing of the current menu and returns to the Primary Options Menu. During Run Mismatch processing, this command is treated as the End command.

Size Def/Alt

Directs CA Verify for CICS to use the default or alternate screen size for its menus. Both the default and alternate sizes are defined during installation. The sizes supported are dependent on terminal models. If this command is specified without either parameter, CA Verify for CICS toggles between the default and alternate sizes.

Suspend

Suspends current processing and displays the Primary Options Menu. The suspended function appears on the menu, followed by (SUSPENDED). Only one Suspend command may be issued from a terminal. This command is not valid from the Primary Options Menu. See Resume.

The following chart lists the commands for which default PF keys are assigned. Default PF key assignments are determined when CA Verify for CICS is installed. Minimum abbreviations are underlined.

Default PF Key	Command
PF1	Help
PF2	Rotate
PF3	End
PF4	Return
PF5	Prev
PF6	Next
PF7	Up
PF8	Down
PF9	Format
PF10	Left
PF11	Right
PF12	Keys

Note: Defaults for PF keys 13-24 are the same as for PF keys 1-12.

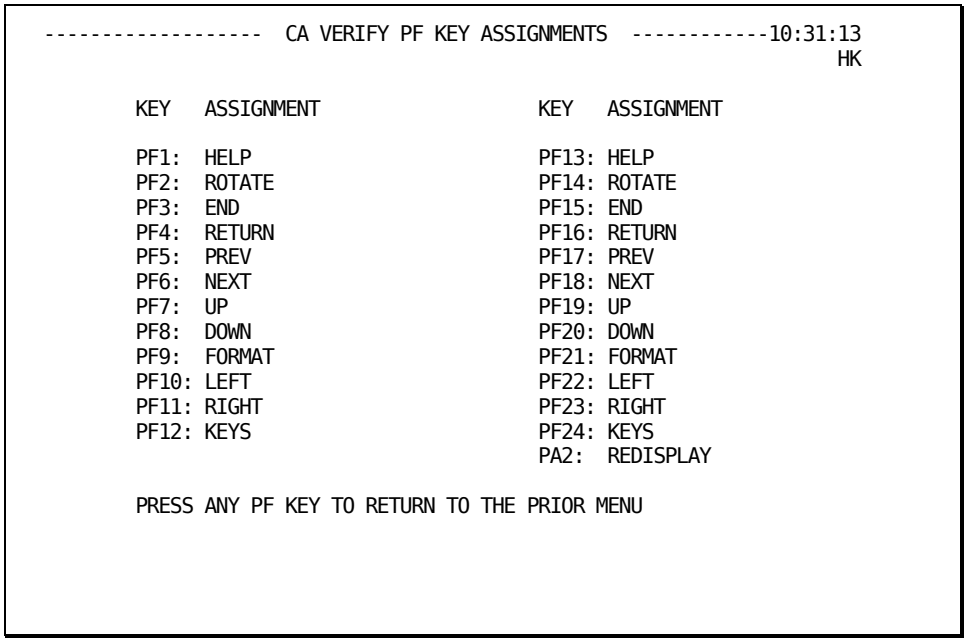
See the preceding section for an explanation of commands used with all functions. Commands for specific functions are discussed in the appropriate chapters.

Be aware of the following:

- If a PF key command is not appropriate to the function you are performing, CA Verify for CICS treats it as the Enter key.
- If a command requires parameters, type them in the command area.
- If you type more parameters than a command requires, CA Verify for CICS ignores them.

Display PF Key Assignments

Type the Keys (PF12) command on any menu to display a list of PF key assignments.



```
------------------  CA VERIFY PF KEY ASSIGNMENTS  ------------10:31:13
                                                                    HK

        KEY   ASSIGNMENT                  KEY   ASSIGNMENT

        PF1:  HELP                        PF13: HELP
        PF2:  ROTATE                      PF14: ROTATE
        PF3:  END                         PF15: END
        PF4:  RETURN                      PF16: RETURN
        PF5:  PREV                        PF17: PREV
        PF6:  NEXT                        PF18: NEXT
        PF7:  UP                          PF19: UP
        PF8:  DOWN                        PF20: DOWN
        PF9:  FORMAT                      PF21: FORMAT
        PF10: LEFT                        PF22: LEFT
        PF11: RIGHT                       PF23: RIGHT
        PF12: KEYS                        PF24: KEYS
                                          PA2:  REDISPLAY

        PRESS ANY PF KEY TO RETURN TO THE PRIOR MENU
```

Chapter 3: Testing

This section contains the following topics:

CICS Testing

Testing CICS programs without CA Verify for CICS is difficult because your test region differs significantly from your production environment. For example, it's very hard to replicate the:

- Number of terminals
- Types of terminals
- Transaction activity

With CA Verify for CICS, it's easy to simulate production conditions. That means you can perform all types of testing — unit, integration, concurrency, regression, stress, and migration — so you know your application will perform as expected. The rest of this chapter will explain in detail how to use CA Verify for CICS to perform these types of testing.

To test applications effectively, you need well-constructed tests. CA Verify for CICS greatly simplifies the task of creating these tests. All you have to do is log test streams. Optionally, you can edit these test streams to meet your testing needs. For example, you can combine test streams, alter the input data to test a variety of conditions, or delete a sequence of screens. As you save test streams, you'll create a library of tests that will always be available to test application and system changes.

To further streamline the testing of application or system changes, CA Verify for CICS provides a powerful Rules function to use with your library of test streams. Rulesets let you identify the expected changes to the existing test stream due to application or system changes. By running the test stream in the new application with a specified ruleset, you can eliminate the need to stop at the expected changes, greatly streamlining your testing.

Unit Testing

Unit testing is the most common type of testing. You change individual *units of work* and test each change. A unit of work may be a program, a sequence of events, or just an input/output screen. However, unit testing does not test how a program interacts with other programs or its effect on CICS.

Follow these steps:

1. Create a test stream by logging the screens connected with the change. If the program is small, you may want to log all its screens.

   If you already have a test stream that tests this unit of work, you won't need to create a new one.

2. Change the screen in the application.

3. Run the logged test stream. CA Verify for CICS compares the output produced by the program *before* the change with the output produced after the change.

   CA Verify for CICS flags all differences. You expect *one* difference — the difference in the single field, which you changed. Otherwise, the original and current screens should be the same.

4. If you find any unexpected differences, make corrections and rerun the logged test stream until you are sure the modified program is functioning as anticipated.

Integration Testing

Integration testing determines if a program works properly with other programs. A program can pass unit testing — and then fail when executed in conjunction with other programs that were not part of the unit test.

Follow these steps:

1. Create a test stream logging all affected programs — that is, the program that changed and all programs run in conjunction with it. Or, use a previously logged test stream that tests this application.

   The logged test stream can be created by one of the following methods:

   - Capturing a series of transactions in your production region
   - Combining several unit test streams. Use the Append or Insert utilities if you want the test streams to run sequentially, or the Merge utility if you want the test streams to run concurrently.

2. Change the screen in the application.

3. After the changed program passes unit testing, run the logged test stream. CA Verify for CICS compares the output produced by all the programs before the change with the output produced after the change.

CA Verify for CICS automatically flags all differences. You expect one difference — the difference in the single field, which you changed. Otherwise, all the original and current screens should be the same.

4. If you find any unexpected differences, make corrections and rerun the logged test stream until you are sure the modified program and all programs connected to it are functioning as anticipated.

Repetition Testing

Another form of integration testing is *repetition* testing. This type of testing ensures that programs remain re-usable and is generally used to test re-entrancy. To perform repetition testing, simply re-execute a test stream a few times to make sure that each execution is successful and produces the same results.

Concurrency Testing

Concurrency testing determines what happens when similar or identical transactions execute at the same time and try to perform the same task, such as processing the same file or data base record.

Manual concurrency testing is virtually impossible. You would have to have multiple users simultaneously enter the same transaction on different terminals. Even if that were practical, network and access method processing would affect how CICS processes the transactions. With CA Verify for CICS, however, concurrency testing is both easy and accurate because CA Verify for CICS automatically ensures that the transactions are processed concurrently.

Follow these steps:

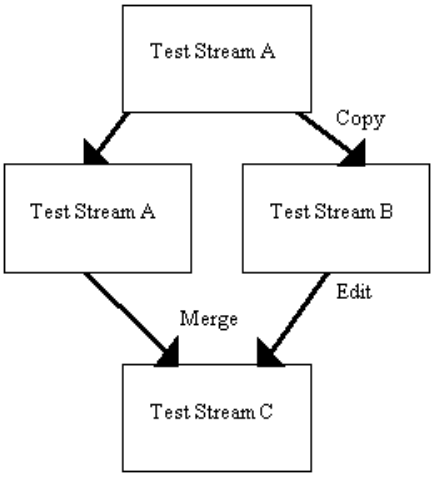
1. Create a test stream by logging the screens when a transaction updates an inventory file. Or, use an existing test stream.
2. Merge the test stream with itself. The new test stream consists of two identical sets of records, each assigned to a different terminal. If necessary, you can merge the new test stream with itself again, creating four identical sets of records assigned to four terminals.
3. Optionally, you can edit the test stream to alter the test data. For example, you may want to alter the inventory file key to ensure that the test stream updates different records. You can edit the test stream by copying it before the merge, editing one test stream, and then merging the two test streams. Or, you can first create the merged test stream and then edit it.

4. Run the merged test stream.
   - CA Verify for CICS sends the first input screen from terminal 1 to the application. Then, CA Verify for CICS sends the first input screen from terminal 2 to the application. These are, of course, identical screens unless you edited one.
   - Next, the application sends the first output screen from the application to terminal 1. Then, the application sends the first output screen from the application to terminal 2.
   - CA Verify for CICS continues to run the test stream. You can easily determine how the application concurrently processes these requests.

When CA Verify for CICS runs a single terminal test stream that has been merged into itself, it ensures that all output screens are received by the terminals before it sends the next input screens. Using this method, a multiple terminal test stream always produces consistent results.

The following diagrams illustrate two ways in which you can edit the test stream.

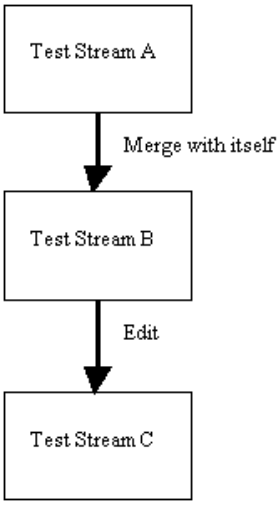
Example 1



In Example 1:

1. The test stream is copied.
2. One of the copies is edited.
3. The two test streams are merged.

Example 2



In Example 2:

1. The test stream is merged with itself.
2. The merged test stream is edited.

Regression Testing

Regression testing ensures that a change to a component of a system does not have unexpected effects on the rest of the system.

For regression testing, you create a benchmark test and then re-execute the system *using the same test data.* CA Verify for CICS compares the results with the benchmark results to determine if there are any unexpected differences.

With CA Verify for CICS, it's easy to create and maintain standardized tests for all your systems. Whenever you make a change, you can quickly determine whether or not the system still performs as expected.

Follow these steps:

1. Create a test stream by logging a standard set of payroll transactions, or use an existing test stream. This test stream will be your benchmark test for the payroll system.
2. Change the program.
3. Unit test the program to make sure the change is working as expected.
4. Run the logged payroll test stream. The only difference CA Verify for CICS should detect is the single program change.

Stress Testing

Stress testing lets you find out how your system behaves under heavy loads and how increased transaction volume affects response time. System programmers can use this information to tune the system. Capacity planners can use it to determine when and how to expand system capability to meet projected growth estimates. Because CA Verify for CICS uses *virtual* terminals instead of real terminals, you can simulate system activity without tying up valuable system resources.

Follow these steps:

1. Create one or more multiple terminal test streams by logging many terminals at once. For example, you may want to log all the terminals in a particular department or all the terminals in a region.
2. Create a large-volume test stream by merging the test stream with itself or by merging several test streams. The screens from one test stream are interspersed among the screens from a second test stream.
3. Run the merged test stream either online or as a batch job.
   - When CA Verify for CICS runs in batch, it brings up a CICS region in which it is the only user. Because there is no resource contention, transactions execute under optimal conditions. Such information may be valuable for optimization or tuning.
   - To simulate increased stress, reduce the operator think time. For example, you can request that input screens be sent to the application twice as quickly as the operators originally sent them. You can even specify no operator think time. Reducing think time is a good way to simulate a stress test even with moderate transaction activity.
   - Because you're not concerned with program changes, instruct CA Verify for CICS *not* to compare the original and current screens. Of course, if you logged only input screens, no comparison will take place.

Note: To test an incremental increase in transaction volume, run a test stream containing the additional volume during peak periods.

Migration Testing

Migration testing ensures that existing applications perform as expected when you have major hardware or software changes; for example, upgrading from one release of CICS to another, adding disk packs, or even migrating from VSE to MVS.

Follow these steps:

1. Create a test stream by logging several critical hours of activity for as many terminals as you believe are necessary to provide a realistic "slice of life." For example, you may want to log all the terminals in a particular branch or department. Or, you may want to create a controlled test by combining and/or editing many unit tests.

   Remember, once you create this test stream, you can use it again and again to test system or software upgrades.

2. At night or on a weekend, run the logged test stream, either from a terminal or as a batch job. Specify no operator think time so the test completes as quickly as possible.

3. If the transactions execute normally and the output is correct, you can assume that the hardware or software change won't have any unexpected impact on production.

Chapter 4: Log Function

This section contains the following topics:

Overview

The Log function is used to create a test stream of input and output terminal screens for an application or system. Once a test stream has been logged, it can be run and rerun to test changes to the underlying program.

For example, suppose you modify an application and want to check for changes and errors in the application that may have been unintentionally introduced. You can log a test stream on the original program and run that test stream on the new version. CA Verify for CICS will compare the logged output screens with the new output screens and flag any differences, making it easy to spot and correct errors. Alternatively, if you install new disk drives or a new release of CICS, you can run your logged test streams to make sure all your applications perform as expected.

You can log a test stream from one or more terminals. Logging a test stream from multiple terminals provides the tools you need for stress testing. You can also run such a test stream under varying conditions to gauge the impact on your system.

As you log test streams, you'll be creating a library of benchmark tests you can use over and over again to test application and system changes. For this reason, you'll want to make sure the test streams you create are as complete as possible.

Note: Test streams can be browsed and edited. For example, you can change a test stream so you can test different input, or you can merge test streams to create a new test case.

How the Log Function Works

Follow these steps:

1. You initiate logging.
2. The next *input* from each terminal being logged will be the *first* screen for that terminal in the test stream. Logging *always* begins with an input screen for each terminal, followed by input and output screens in any order.
3. Logging continues until you explicitly terminate it

CA Verify for CICS can log multiple test streams at the same time. These test streams can be logged from the same or different terminals. For example, two users can —at the same time—request that terminal A be logged. CA Verify for CICS would then create two test streams.

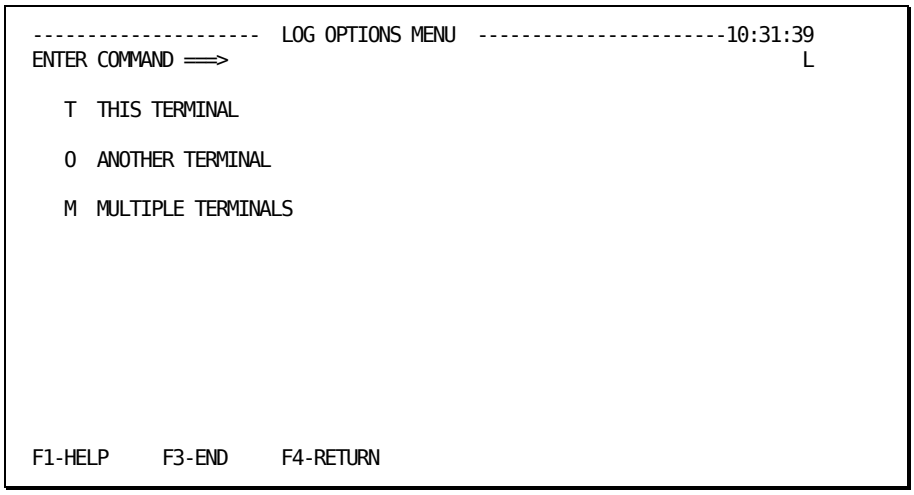
You won't even be aware that logging is taking place. CA Verify for CICS does *not* tie up a terminal that is being logged. A single control task, which does *not* use a terminal, records the screens for *all* logging.

Invoke the Log Function and Specify Terminal Options

Follow these steps:

1. Type **L** on the Primary Options Menu.

   CA Verify for CICS displays the Log Options Menu.



2. Indicate which terminals CA Verify for CICS should log by specifying:

   T

   Your terminal

   O

   Another terminal

   M

   Multiple terminals

3. Press Type.

   CA Verify for CICS displays the menu for the option you have specified.

   Note: You can bypass the Log Options Menu by typing L.T, L.O, or L.M on the Primary Options Menu.

Log One or More Terminals

You have the option of logging a test stream on the terminal on which you are operating CA Verify, on another terminal, or on multiple terminals.

Follow these steps:

1. If you specify **T** on the Log Options Menu or **L.T** on the Primary Options Menu, CA Verify for CICS displays the Single Terminal Log menu. You can also display this menu typing **XTCA L.T** when you invoke CA Verify for CICS from CICS.

```
------------------------  SINGLE TERMINAL LOG  --------      --------13:45:18
ENTER COMMAND ===>                                                      L1

LOG TEST STREAM AS:
  DDNAME      ===> TCADS
  APPLICATION ===>
  MEMBER      ===>
  VERSION     ===> 001

DESCRIPTION ===>
            ===>
            ===>

STOP OPTION                    ===> man  (MAN, PF__, PA_, OR CLEAR)
PROCESS WITH RULES             ===> N    (Y/N)
TEST STREAM PROTECTION         ===>      (R-READ W-WRITE P-PRINT)
LOG INPUT SCREENS ONLY         ===> N    (Y/N)
EXTEND TEST STREAM             ===> N    (Y/N)


RULESET NAME:


F1-HELP     F3-END     F4-RETURN
```

VSE User Note:

Test streams under MVS specify ddname fields, while test streams under VSE specify FILENAME fields. VSE users should substitute FILENAME for ddname wherever ddname occurs in this guide.

Default values, specified when CA Verify for CICS is installed, will appear for the following fields: ddname, Application, Version, Stop Option, and Extend Test Stream.

The ddname, Application, Member, Version, and Description fields are required to identify the test stream.

DDNAME

The ddname of the file to which the test stream will be logged. Specify one to eight alphanumeric or national characters. The first character cannot be numeric. This file must be allocated in the JCL for CICS.

APPLICATION

The group to which the test stream belongs; e.g., an application, department, project, etc. Specify one to eight alphanumeric or national characters. The first character cannot be numeric.

MEMBER

The specific test stream; e.g., a screen or program. Specify one to eight alphanumeric or national characters. The first character cannot be numeric.

VERSION

The test stream version when multiple versions of the test stream have the same name. Specify one to three numeric characters.

DESCRIPTION

The comments that describe the test stream. Specify up to three lines of information. You may want to use this field to document files which must be restored when the test stream is run. This field is ignored when the Extend Test Stream option is Y.

STOP OPTION

The option which will terminate logging:

MAN

Logging terminates when you manually stop logging in one of two ways:

- Clear your screen and type **XTCA STOP**
- Use the Inquiry function to display a list of all active logging and select the logging you wish to terminate. See the Inquiry function (see page 205) chapter for more details.

  Only the person who invoked logging or a security administrator can terminate logging using the Inquiry function.

  Note: The MAN option can always be used to stop logging even if you select another Stop option.

PFxx/Pax

Logging terminates when you press the specified PF or PA key.

CLEAR

Logging terminates when you clear your screen.

PROCESS WITH RULES

This field allows you to apply a ruleset to this logging session. If **Y** is typed in this field, a list of available rulesets will appear when you press Enter. These rulesets contain rules that specify user IDs, terminals, or transaction IDs that are to be included or excluded from the test stream. If you do not wish to apply a ruleset to this logging session, leave **N** as the default.

Note: L.T Only applies transaction rule actions; terminal and user rule actions are ignored.

TEST STREAM PROTECTION

Sets protection options for the test stream. Valid options are:

R

Read: the test stream cannot be browsed, edited, run, appended to, inserted into, or merged into another test stream by anyone *except* its owner or a security administrator. If the test stream is copied or renamed, read protection is extended to the new test stream.

W

Write: the test stream cannot be extended, replaced (during the Run function), edited, renamed, deleted, appended to, inserted or merged into another test stream, nor can its directory be updated, by anyone *except* its owner or a security administrator.

P

Print: the test stream cannot be printed. You can use any combination of options. For example., RW, WP, RWP. To change the protection option, use the Utilities Update Directory function.

Note: A read-protected Autsign (automated signon and signoff) test stream can be invoked by any user for Autsign purposes; it cannot be read in any other way except by its owner or a security administrator. For information about Autsign test streams, see the *Installation Guide*.

LOG INPUT SCREENS ONLY

Type Y to log just input screens; otherwise, leave N as the default. Use Y with caution, because it means there will be no output screens to compare during a Run. Selecting Y also diminishes the usefulness of CA Verify for CICS as a regression testing tool, and makes problem determination difficult if the application does not perform as expected. However, when you have a completely predictable test stream, logging just the input screens can reduce the size of the test stream.

EXTEND TEST STREAM

Type **Y** to append additional logging to an existing test stream (identified in the ddname, Application, Member, and Version fields); otherwise, leave **N** as the default.

If you extend a test stream, the current logging should be compatible with the existing test stream as follows:

- The current terminal should be the same type as the terminal originally used to create the test stream
- The existing test stream should end with a clear screen. If not, use the Online Utilities to copy the existing test stream. You can then truncate the copied version after a clear screen.

RULESET NAME

Displays the name of the ruleset selected. A name only appears if you have selected Y for Process With Rules and have selected a ruleset.

Initiate Logging on Your Terminal

When you have completed this menu, press Enter.

- If you entered **N** to Process With Rules, CA Verify for CICS clears your screen and begins logging at your terminal.
- If you entered **Y**, the next screen to display will be the Specify Ruleset Name panel.

See Apply Rules to the Logging Session in this chapter for details.

Log Another Terminal

If you enter **O** on the Log Options Menu or **L.O** on the Primary Options Menu, CA Verify for CICS displays the Other Terminal Log menu. You can also display this menu by typing XTCA L.O when you invoke CA Verify for CICS from CICS.

```
---------------------------  OTHER TERMINAL LOG ---------------10:32:18
ENTER COMMAND ===>                                                  L2

LOG TEST STREAM AS:
  DDNAME      ===> TCADS
  APPLICATION ===>
  MEMBER      ===>
  VERSION     ===> 001

DESCRIPTION   ===>
              ===>
              ===>

TERMINAL TO BE LOGGED     ===>
STOP OPTION               ===> MAN         (MAN, PF__, PA_, OR CLEAR)
PROCESS WITH RULES        ===> N           (Y/N)
TEST STREAM PROTECTION    ===>             (R-READ W-WRITE P-PRINT)
LOG INPUT SCREENS ONLY    ===> N           (Y/N)
EXTEND TEST STREAM        ===> N           (Y/N)



RULESET NAME:


F1 - HELP     F3 - END        F4 – RETURN
```

This menu is identical to the Single Terminal Log menu except for one additional field, Terminal to be Logged. Also, the Stop option is interpreted differently and the Proces With Rules field only applies certain rule actions.

TERMINAL TO BE LOGGED

The name of the terminal to be logged. CA Verify assumes this is the terminal netname *unless* a field mark (X'1E') follows the name. In this case, CA Verify for CICS assumes the name is the CICS terminal ID. The field mark key is labeled Field Mark or FM on most keyboards.

STOP OPTION

How logging will be terminated. See the Log One or More Terminals (see page 40) section for an explanation of the options.

Remember that the Stop option you select applies to the terminal being logged, *not* to your terminal. For example, if you select PF12, the operator of the terminal being logged must press PF12 to terminate logging. If the operator at that terminal is unaware of the logging, select MAN as the Stop option. Then you can use the Inquiry function to terminate logging.

PROCESS WITH RULES

If **Y** is typed in this field and **L.O** is specified, transaction and user rule actions are applied, while terminal rule actions are ignored.

Initiate Logging on Another Terminal

When you have completed this menu, press Enter.

- If you typed **N** to Process With Rules, CA Verify for CICS redisplays the Log Options Menu and informs you that logging has begun.
- If you typed **Y**, the next screen to display will be the Specify Ruleset Name panel.

See Apply Rules to the Logging Session in this chapter for details.

Log Multiple Terminals

If you type **M** on the Log Options Menu or **L.M** on the Primary Options Menu, CA Verify for CICS displays the Multiple Terminal Log menu. You can also display this menu by typng **XTCA L.M** when you invoke CA Verify for CICS from CICS.

```
--------------------  MULTIPLE TERMINAL LOG  -------------------10:32:36
ENTER COMMAND ===>                                                  L3
LOG TEST STREAM AS:
  DDNAME      ===> TCADS
  APPLICATION ===>
  MEMBER      ===>
  VERSION     ===> 001

DESCRIPTION   ===>
              ===>
              ===>

PROCESS WITH RULES             ===> Y          (Y/N)
TEST STREAM PROTECTION         ===>            (R-READ W-WRITE P-PRINT)
LOG INPUT SCREENS ONLY         ===> N          (Y/N)


RULESET NAME:



F1 - HELP     F3 - END        F4 – RETURN
```

This menu is similar to the Single Terminal Log and Other Terminal Log menus with these exceptions:

- There is no Stop option because you must terminate logging *manually* via the Inquiry function.
- There is no Extend Test Stream option because multiple terminal test streams cannot be extended. Use the Append utility to extend multiple test streams (see the "Utilities (see page 211)" chapter).
- The Process With Rules option is *automatically* set to Y. The terminals to be logged *must* be listed in a LOGGING ruleset. See Apply Rules to the Logging Session in this chapter for instructions.

  Note: L.M applies all three rule actions: transaction, user, and terminal.

When you have completed this menu, press Enter to display the Specify Ruleset Name panel. See Apply Rules to the Logging Session in this chapter for instructions on completing this panel.

Multiple Terminal Logging Considerations

When logging multiple terminals:

- The terminals to be logged *must* be listed in the LOGGING ruleset. See Add a Multiple Terminal Logging Rule Action *(see page 54)* in this chapter.
- Bear in mind that at least 8 KB per terminal for the Run function and 2 KB to 8 KB for the Print and Browse functions are required. You can reduce the Run requirements to 100 bytes per terminal by using the Batch Run function or the online Run function with the Stop at Mismatches option set to N.
- CA Verify for CICS tries to ensure that the same concurrences exist when test streams are run as when they were logged. However, timing considerations could possibly cause problems because CA Verify for CICS runs test streams so much faster than when they were logged.

  For example, suppose an operator at terminal A adds an item to a file. Next, an operator at terminal B inquires on the same item. If the inquiry occurs *after* a confirmation message has been sent to terminal A, CA Verify for CICS guarantees the correct sequence during the Run function. However, if the inquiry occurs *before* the confirmation message has been sent, during the Run function the inquiry may actually occur *before* the item has been added.

CICS/ESA FEPI Considerations

CA Verify for CICS is capable of logging and running CICS/ESA FEPI front-end and back-end applications. However, because the CICS/ESA FEPI option uses its own type of virtual terminals to drive the back-end applications, you must be aware of the following considerations:

- You must not log both real (front-end) terminals and FEPI virtual (back-end) terminals in the same multiple terminal test stream or the results from a subsequent RUN function will be unpredictable. This is because when the RUN function executes the FEPI front-end application(s), the FEPI back-end applications are automatically re-executed at the same time CA Verify for CICS is also trying to execute the same back-end applications.
- You can log multiple terminal test streams for either FEPI front-end applications or back-end applications in different test streams. When you subsequently RUN the front-end application test stream, both the front-end and back-end applications will be executed, and when you subsequently RUN the FEPI back-end test stream, only the back-end applications are executed.
- For CA Verify EEO only, the same previous considerations apply when logging and running test-streams using the L.A option.

Apply Rules to the Logging Session

Before initiating a logging session, you can choose to apply a set of rules to this session. Logging rules allow you to specify a list of user IDs, terminals, or transaction IDs that are to be included or excluded from the test stream. Logging rules can be created in system, application, or test stream rulesets.

Important! You cannot add a new **ruleset** when you are setting up for the Log function. You must create the ruleset using the Maintain Rules selection from the Primary Options menu before performing the log session.

To apply a ruleset to the logging session, type **Y** in the Process With Rules field on the Terminal Log Screen, as explained in the section Logging One or More Terminals. When you complete the logging information on the screen and press Enter, you will be prompted to select or specify the name of the ruleset to apply to this session.

Note: There is a hierarchy in applying rulesets. During logging, system rulesets are applied first, application next, and test stream rulesets last. During a run, test stream rulesets are applied first, application next, and system rulesets last.

Specify a Ruleset to Apply

The first screen to appear when you press Enter on the Terminal Log screen is the Log: Specify Ruleset Name panel, shown following. Application and system rulesets do not need to be specified during a logging session; if they exist, they are automatically applied. You also cannot exclude application and system rulesets from a logging session.

```
------------------------- LOG: SPECIFY RULESET NAME------------------15:53:10
 ENTER COMMAND ===>

S-SELECT    B-BROWSE    E-EDIT

 _ TEST STREAM RULESET
   DDNAME      ===> TCADS
   APPLICATION ===>                (LEAVE APPLICATION, MEMBER,
   MEMBER      ===>                 OR VERSION BLANK AND PRESS
   VERSION     ===>                 ENTER FOR A SELECTION LIST)

 APPLICATION RULESET:  TCADS.application.#RULESET.001
   DESCRIPTION:


 SYSTEM RULESET:  TCADS.SYSTEM.#RULESET.001
   DESCRIPTION:



F1-HELP      F3-END       F4-RETURN
```

Rulesets are identified by DDname, Application, Member, and Version information. If you know the name of the ruleset you want to apply, type the information in the Test Stream Ruleset fields. If you do not know the name of the ruleset, see the next section (see page 48) for instructions on selecting a ruleset from a list.

When you are finished, type **S** to the left of the ruleset name and press Enter. You will return to the Terminal Log panel and the information you specified will appear in the Ruleset Name field. To initiate logging, press Enter from this panel.

Edit the Ruleset

To view or modify your specified ruleset before selecting it to apply, type the information, type **B** to browse or **E** to edit to the left of the ruleset and press Enter. The Edit Rules—Summary screen will appear with the details of your selected ruleset. See the section Add or Edit a Logging Rule *(see page 50)* for information on how to edit a ruleset or add a rule from this screen.

Note: You cannot create a new ruleset while using the Log function; use the Rules function to create the new ruleset first.

Select a Ruleset From a List

If you do not know the exact name of the ruleset you want to apply, leave the ruleset information fields on the Specify Ruleset Name panel blank, type B to the left of the ruleset, and press Enter. The Browse: Ruleset Selection panel will appear, displaying a list of available rulesets. You can select a ruleset from the list and return to the Specify Ruleset Name panel.

```
------------------------- BROWSE: RULESET SELECTION ------------------14:07:32
 ENTER COMMAND ===>
 FILE: TCADS
 S-SELECT

                          --CREATED----BY---- --UPDATED----BY----  RULES ITEMS
 _ ITEST    TEST     001   04/22/1998          04/22/1998              1     1
                     TEST DATAGEN
 _ LOGRULES SAMPLE   001   05/19/1998          06/19/1998              1     3
                     A BUNCH OF LOGGING RULES
 _ LOGRULES SAMPLE   004   06/19/1998          06/19/1998              1     1
                     JUST TERM RULE ACTION
 s LOGRULES SAMPLE   005   06/19/1998 GRIDO01  07/07/1998              1     2
                     MULT TERM ALL ='S; COUPLE USERID'S
 _ SAMPLE   DADS     001   10/30/1997          11/04/1997              4    26
                     SAMPLE RULESET USING MODELLING
                     ** KEEP THIS !! DO NOT DELETE !!  **
 _ SAMPLE   DADS     999   05/19/1998          05/19/1998              3     5
                     SAMPLE OF DADS 3.4 TEST STREAM
 _ SYSTEM   #RULESET 001   04/14/1998          05/29/1998              1     1
                     SYSTEM RULESET
 _ SYSTEM   #RULESET 998   05/19/1998                                  1     1
```

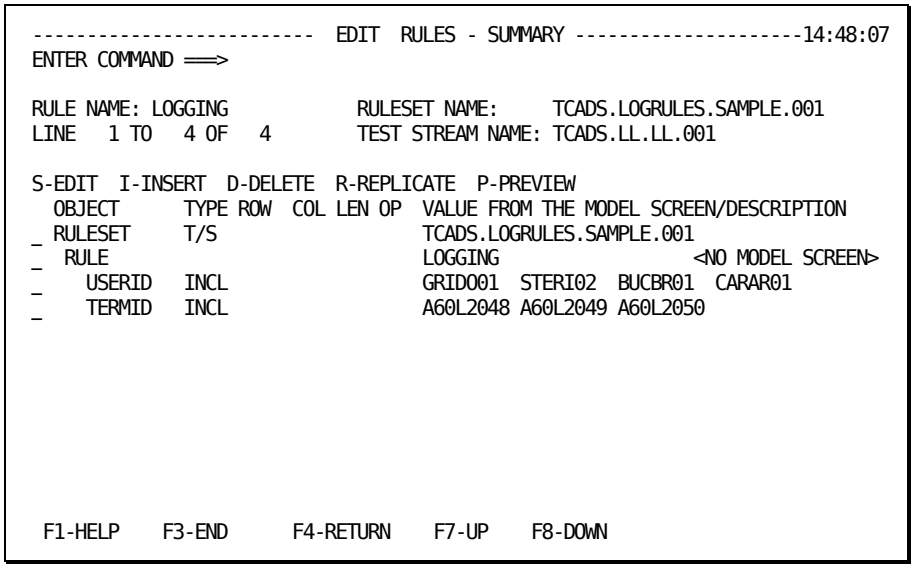
This panel lists all defined rulesets in alphabetical order by ruleset name. The information shown for each ruleset includes: member, version, date the ruleset was created and who created it, the date it was updated and who updated it, the number of rules the ruleset contains, and the number of rule actions. A description of the ruleset is displayed on the line(s) beneath the ruleset name.

To select a ruleset to be applied to this logging session, type S next to the ruleset you are selecting as shown in the previous example. Press Enter to return to your log screen (either single, other, or multiple terminal). Press Enter to begin logging.

Add or Edit a Logging Rule

Follow these steps:

1. Select a ruleset from the Ruleset Selection panel by typing **S** to the left of the ruleset name that you want to edit
2. Press Enter. The Log: Specify Ruleset Name panel will redisplay with the name of the ruleset you selected placed in the TEST STREAM RULESET field.
3. Type **E** next to TEST STREAM RULESET and press Enter. The Edit Rules—Summary panel will appear as shown following:



```
-------------------------  EDIT  RULES - SUMMARY --------------------14:48:07
ENTER COMMAND ===>

RULE NAME: LOGGING           RULESET NAME:     TCADS.LOGRULES.SAMPLE.001
LINE   1 TO   4 OF   4       TEST STREAM NAME: TCADS.LL.LL.001

S-EDIT  I-INSERT  D-DELETE  R-REPLICATE  P-PREVIEW
  OBJECT      TYPE ROW  COL LEN OP  VALUE FROM THE MODEL SCREEN/DESCRIPTION
_ RULESET     T/S                   TCADS.LOGRULES.SAMPLE.001
_  RULE                             LOGGING                 <NO MODEL SCREEN>
_    USERID   INCL                  GRID001  STERI02  BUCBR01  CARAR01
_    TERMID   INCL                  A60L2048 A60L2049 A60L2050




 F1-HELP    F3-END      F4-RETURN    F7-UP    F8-DOWN
```

The Edit Rules—Summary panel summarizes the rules contained in your selected ruleset. It includes the following information.

OBJECT

This column lists the rules within this ruleset and the rule actions within each rule. See the chapter "Rules Function (see page 151)" for a list of values that can appear in the OBJECT column.

TYPE

Indicates what type of ruleset or rule this is. See the chapter "Rules Function (see page 63)" for a list of values that can appear in this column.

The column values for ROW, COL, LEN, OP and VALUE are taken from the panel where that particular rule action was defined.

The actions you can take from this panel include:

- Adding a logging rule or logging rule action. Type **I** next to an existing rule and press Enter. The Add Rules—Rule Action panel appears.
- Editing the description of the ruleset. Type **S** to the left of the ruleset and press Enter. The Edit Rules—Ruleset Description panel appears. See the chapter "Rules Function (see page 63)" for detailed instructions on how to edit a ruleset's description.
- Editing the rule. Type **S** to the left of the rule to be edited and press Enter. See the chapter "Rules Function (see page 151)" for detailed instructions on how to edit a rule.
- Editing a logging rule action. Type **S** to the left of the rule action and press Enter. An Edit Rule Action screen appeas where you can edit the information for the rule action you selected.

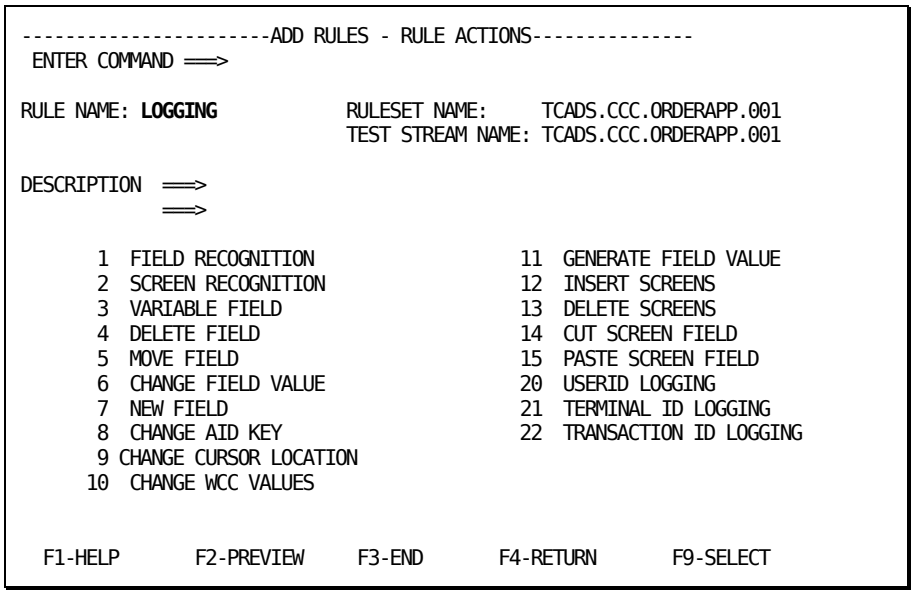
Each logging-specific action is discussed in one of the following sections.

Add a Logging Rule and a Logging Rule Action

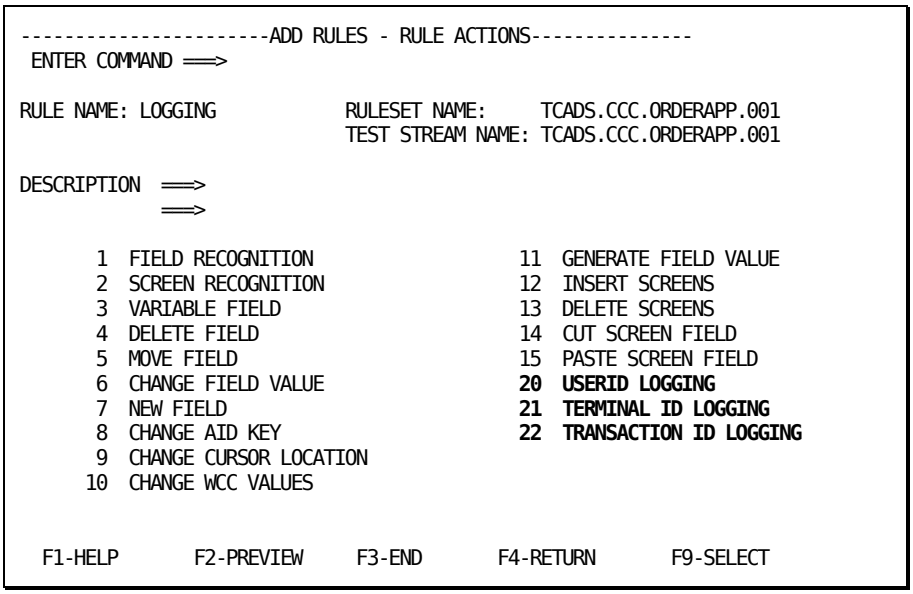
Follow these steps:

1. Start at the Edit Rules—Summary panel.
2. Type **I** next to an existing rule and press Enter to display the Add Rules—Rule Action panel.
3. Enter **LOGGING** as the name of the rule. The name LOGGING is required; entering a description is optional.

   Note: There can only be one rule named LOGGING in a ruleset.



4. Select the appropriate rule action: either 20 (Userid Logging), 21 (Terminal ID Logging), or 22 (Transaction ID Logging).



Use selections 20 and 22 to create logging rule actions to control the user IDs (USERID) or the transaction IDs (TRANID) that are to be included or excluded from the logging session. Use selection 21 to create a rule action that lists the terminal IDs (TERMID) to be included for a multiple terminal logging session.

Selections 20, 21, and 22 are the only rule actions that can be specified within the LOGGING rule. The LOGGING rule can contain one of each of these logging rule action types.

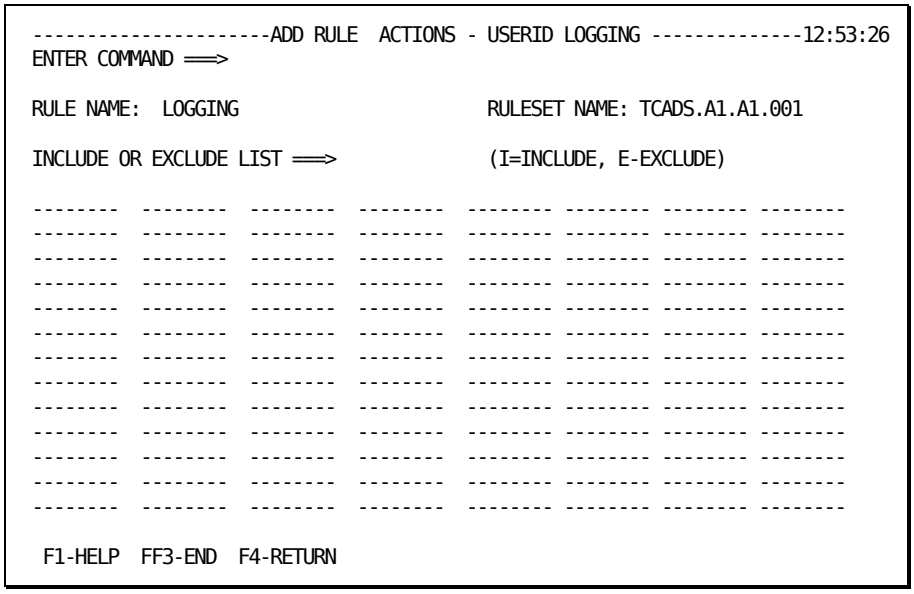
Each type of logging rule action and its associated screen are discussed in the following sections:

Add a User ID Logging Rule Action

Follow these steps:

1. Select option 20 from the Rules menu.

   This displays the Add Rule—Userid Logging screen, as shown following:



```
----------------------ADD RULE  ACTIONS - USERID LOGGING -------------12:53:26
ENTER COMMAND ===>

RULE NAME:  LOGGING                        RULESET NAME: TCADS.A1.A1.001

INCLUDE OR EXCLUDE LIST ===>               (I=INCLUDE, E-EXCLUDE)

--------  --------  --------  --------  -------- -------- -------- --------
--------  --------  --------  --------  -------- -------- -------- --------
--------  --------  --------  --------  -------- -------- -------- --------
--------  --------  --------  --------  -------- -------- -------- --------
--------  --------  --------  --------  -------- -------- -------- --------
--------  --------  --------  --------  -------- -------- -------- --------
--------  --------  --------  --------  -------- -------- -------- --------
--------  --------  --------  --------  -------- -------- -------- --------
--------  --------  --------  --------  -------- -------- -------- --------
--------  --------  --------  --------  -------- -------- -------- --------
--------  --------  --------  --------  -------- -------- -------- --------
--------  --------  --------  --------  -------- -------- -------- --------

 F1-HELP  FF3-END  F4-RETURN
```

2. Indicate whether this is a list of user IDs to be included or excluded from the logging session.
3. In the Include or Exclude List field, enter an **I** for include or an **E** for exclude.

   Note: Depending on the security settings, user IDs that are excluded from the logging session may still view the log.

4. Enter up to 120 specific or generic user IDs in the fields provided.

   The user ID must exactly match the one maintained in the security system (it is case sensitive and there must be no trailing underscore).

5. Press Enter to add the information to the rule.
6. Press PF3 to return to the Edit Rules—Rules Summary panel.

Use an Exclude List With a Terminal ID Include List

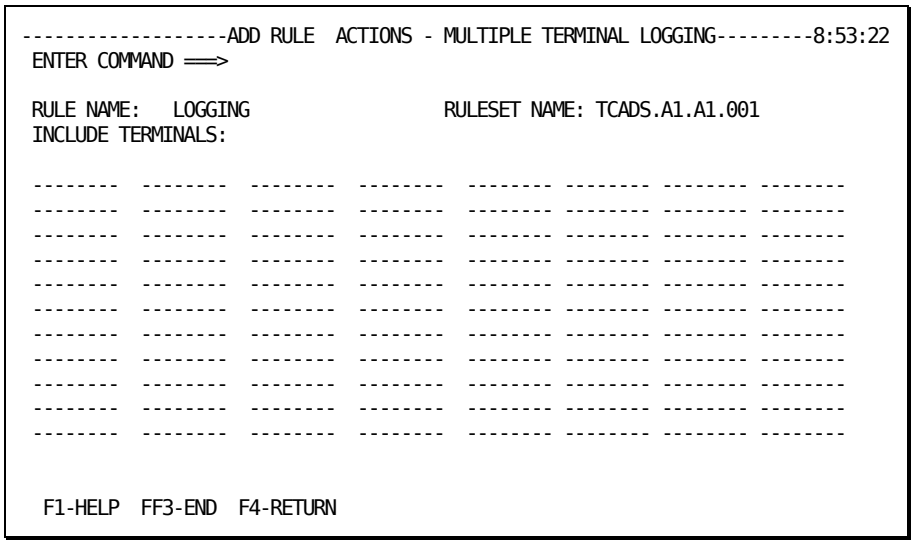
When a user ID logging rule action *and* a multiple terminal logging rule action are specified in the same ruleset, an exclusion will always take precedence over an inclusion. For example, USER01 and USER02 are listed in the exclude list of a user ID logging rule action, while USER03 is not (therefore, USER03 will be included in the logging session). TERM01 and TERM02 are in the list of terminals to be included in a multiple terminal logging rule action, while TERM03 is not (therefore, TERM03 will not be included in the logging session). If USER01 (excluded) is working on TERM01 (included), USER01 will not be included in the logging session. Likewise, if USER03 (included) is working on TERM03 (excluded), then USER03 will not be included in the logging session.

Add a Multiple Terminal Logging Rule Action

Follow these steps:

1. Select option 21.

   A sample screen is shown following:



2. Type up to 120 specific or generic terminal IDs to be included in the logging session in the fields provided.

   The terminal ID must exacly match the one maintained in the TCTIE (it is case sensitive). You can use = as a wildcard to replace any character. For example, if you instruct CA Verify for CICS to log TRM==, CA Verify for CICS will log all terminals beginning with the characters TRM followed by up to two characters; such as, TRM1, TRM01, TRM29, and so on.

Note: CA Verify for CICS assumes the terminal name is its netname *unless* a field mark (X'1E') follows the name. In this case, CA Verify for CICS assumes the name is the CICS terminal ID. The field mark key is labeled Field Mark or FM on most keyboards.

3. When you are finished, press Enter to add the information to the rule; press PF3 to return to the Rules Menu.

Use an Exclude List with a Terminal ID Include List

When a user ID logging rule action or a transaction ID logging rule action **and** a multiple terminal logging rule action are specified in the same ruleset, an exclusion will always take precedence over an inclusion. For example, USER01 and USER02 are listed in the exclude list of a user ID logging rule action, while USER03 is not (therefore, USER03 will be included in the logging session). TERM01 and TERM02 are in the list of terminals to be included in a multiple terminal logging rule action, while TERM03 is not (therefore, TERM03 will not be included in the logging session). If USER01 (excluded) is working on TERM01 (included), USER01 will not be included in the logging session. Likewise, if USER03 (included) is working on TERM03 (excluded), then USER03 will not be included in the logging session.

Add a Transaction ID Logging Rule Action

Follow these steps:

1. Select option 22.

   A sample screen is shown following:

```
 ------------------ADD RULE  ACTIONS - TRANSACTION ID LOGGING ---------16:58:56
 ENTER COMMAND ===>

 RULE NAME:   LOGGING                    RULESET NAME: TCADS.A1.A1.001
NCLUDE OR EXCLUDE LIST ===>             (I=INCLUDE, E-EXCLUDE)

 --------  --------  --------  --------  -------- -------- -------- --------
 --------  --------  --------  --------  -------- -------- -------- --------
 --------  --------  --------  --------  -------- -------- -------- --------
 --------  --------  --------  --------  -------- -------- -------- --------
 --------  --------  --------  --------  -------- -------- -------- --------
 --------  --------  --------  --------  -------- -------- -------- --------
 --------  --------  --------  --------  -------- -------- -------- --------
 --------  --------  --------  --------  -------- -------- -------- --------
 --------  --------  --------  --------  -------- -------- -------- --------
 --------  --------  --------  --------  -------- -------- -------- --------
 --------  --------  --------  --------  -------- -------- -------- --------
 --------  --------  --------  --------  -------- -------- -------- --------
 --------  --------  --------  --------  -------- -------- -------- --------
 --------  --------  --------  --------  -------- -------- -------- --------
 --------  --------  --------  --------  -------- -------- -------- --------

  F1-HELP  FF3-END  F4-RETURN
```

2. Indicate whether this is a list of transaction IDs to be included or excluded from the logging session.
3. In the Include or Exclude List field, enter an **I** for include or an **E** for exclude. Observe these rules when including or excluding transactions:
   - A transaction that has been globally excluded in the options module cannot be included in the logged test stream.
   - To include or exclude a transaction that has both a TRANSID and TASKREQ in the PCT or RDO definition, specify the 1-4 character transid. For a transaction which has only a TASKREQ in the PCT, specify the PA or PF key. Do not use a high-order zero in the numeric portion of the PA/PF number.
   - Use the transid CSPK to log the print key.

   Note: Including or excluding transactions may cause mismatches or other errors (for example, the logged and run screens may not be synchronized) when you use the Run function.
4. Enter up to 195 specific or generic transaction IDs in the fields provided.

   The transaction ID must exactly match the one maintained in the CSD (it is case sensitive). You can use = as a wildcard to replace any character. For example, if you instruct CA Verify for CICS to include TRN =, CA Verify for CICS will log all screens associated with transactions beginning with the characters TRN followed by one character (for example. TRN1, TRN2 and so on.)
5. When you are finished, press Enter to add the information to the rule.
6. Press PF3 to return to the Rules Menu.

Use an Exclude List with a Terminal ID Include List

When a transaction ID logging rule action and a multiple terminal logging rule action are specified in the same ruleset, an exclusion will always take precedence over an inclusion.

Edit a Logging Rule Action

Follow these steps:

1. Start at the Edit Rules—Summary panel.
2. Tab to the rule action you want to edit.
3. Type **S** and press **Enter** to display the Edit Rule Actions screen for the logging rule action you have selected. For example, if you selected a TRANSID rule action to edit, the Edit Rule Actions—TRANSID Logging screen will appear showing the current definition of this rule action.

4. Add, delete, or overtype any of the current values following the same guidelines used to add this type of rule action.
5. Press **Enter** when you are finished to add the information to the rule. At this point, you can press PF3 to return to the Edit Rules—Summary panel to add or edit another rule in this ruleset or press PF3 to return to the Ruleset Selection panel.

Apply the Ruleset to the Logging Session

Follow these steps:

1. Once the name of the ruleset is entered in the Specify Ruleset Name panel, enter **S** next to the test stream ruleset to select it for the current logging session as shown following:

```
-------------------------SPECIFY RULESET NAME----------------------15:53:10
 ENTER COMMAND ===>

B-BROWSE   E-EDIT           S-SELECT

 S TEST STREAM RULESET
   DDNAME      ===> TCADS
   APPLICATION ===> A1               (LEAVE APPLICATION, MEMBER,
   MEMBER      ===> A1               OR VERSION BLANK AND PRESS
   VERSION     ===> 001              ENTER FOR A SELECTION LIST)

 APPLICATION RULESET:  TCADS.application.#RULESET.001
  DESCRIPTION:


 SYSTEM RULESET:  TCADS.SYSTEM.#RULESET.001
  DESCRIPTION:


F1-HELP      F3-END       F4-RETURN
```

2. Press Enter to return to the Terminal Log panel where you can initiate the logging session.

Initiate Logging at CICS Startup

You may want to begin logging as soon as CICS starts up. One way to do this is for an operator to initiate logging, but it's possible that some transactions may be entered before logging is initiated.

Follow these steps:

1. Log a test stream with the following name:

   DDname

   Default

   Application

   CLIST

   Member

   Meaningful name; for example, LOGSTART

   Version

   Default

2. Include the following in the test stream:

   a. Enter the following:

      XTCA L.M.

   b. Complete the Multiple Terminal Log menu, naming the test stream to be logged; for example, GENERAL.NETLOG.001.

   c. Press **Enter** to initiate logging.

3. Terminate the log of **CLIST.LOGSTART**.

   Now you have a test stream which will initiate logging whenever it is run.

4. Next, add a card reader/line printer to your TCT which contains:

   XTCA EXEC LOGSTART

   This automatically initiates test stream LOGSTART whenever CICS starts up.

5. If you intend to use LOGSTART at each CICS startup, include screens in the beginning to delete or rename the NETLOG test stream from the prior CICS run. Otherwise, LOGSTART will get a *Name in Use* message when it tries to start the log.

6. Alternately, you can edit the LOGSTART test stream and change the version number to activate logging during the next CICS startup. Leaving the version the same can be used to prevent activation. In other words, change the version *only* when you want to activate logging.

See Command List Processor (see page 107) in the "Run Function" chapter for more information.

Terminate the Log Function

CA Verify for CICS stops capturing input and output screens as soon as you request log termination.

For single terminal and other terminal logging, the Stop option determines how logging is terminated. For example, if you specify a PF key, pressing that key will terminate logging.

Multiple terminal logging must be terminated manually via the Inquiry function. You can also terminate single and other terminal logging manually even if you have designated another Stop option.

Follow these steps:

1. Clear your screen
2. Type **XTCA STOP** or access the Primary Options Menu.
3. Select the Inquiry function.

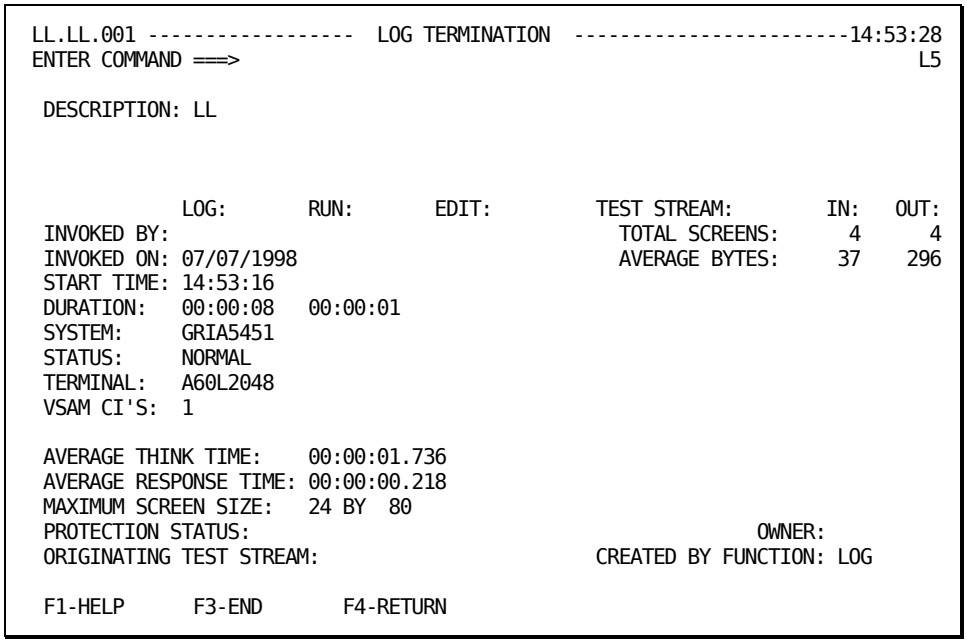
   CA Verify for CICS displays a list of all functions, as illustrated next.

```
-------------------------  INQUIRY/TERMINATION  ----------------------15:03:43
ENTER COMMAND ===>                                                        I1

            INVOKED   AT              USING     RECORD
  FUNCTION  BY        TERMINAL  TIME  TERMINAL  NUMBER  TEST STREAM NAME
_ LOG                 A60L2048  15:03 A60L2048       8  LL.LL.001
_ INQUIRY             A60L2004  15:03

  DATA CAPTURE BUFFERS  00% FULL, NO SECONDARY BUFFERS ALLOCATED









  TYPE AN "S" NEXT TO YOUR LOG FUNCTION TO STOP IT
  TYPE AN "I" NEXT TO A RUN FUNCTION TO INTERRUPT IT (AT ORIGINATING TERMINAL)
F1-HELP       F3-END       F4-RETURN      F7-UP          F8-DOWN
```

4. Type **S** next to the Log function you want to terminate and press Enter.

See the "Inquiry Function (see page 205)" chapter for a detailed explanation of this menu.

When you request log termination, CA Verify for CICS displays the Log Termination menu. If you are logging more than one test stream from the same terminal and you stop them by the same PF key or by entering XTCA STOP, the Log Termination screen will first display information for the most recently initiated test stream.



DESCRIPTION

The test stream description entered when logging was initiated.

INVOKED BY

The ID of the user who initiated logging.

INVOKED ON

The date when logging began.

START TIME

The time when logging began.

DURATION

The time, in hours, minutes, and seconds, which elapsed during logging.

SYSTEM

The CICS jobname.

STATUS

The log completion status:

Normal Incomplete:

CA Verify for CICS could not log some screens because the data capture buffer overflowed

Ext:nnn:

The data stream was extended; nnn indicates the number of records that existed before the extension.

TERMINAL

The name of the logged terminal. For multiple terminal test streams, this field indicates the number of terminals logged.

VSAM CI'S

The number of control intervals used to store the test stream.

AVERAGE THINK TIME

The average time that elapsed between an output screen and the next input from the terminal.

AVERAGE RESPONSE TIME

The average time that elapsed between terminal input and the next output screen.

MAXIMUM SCREEN SIZE

The maximum screen size used during logging.

PROTECTION STATUS

The Test Stream Protection option specified when logging was initiated.

ORIGINATING TEST STREAM

No information appears in this field for a test stream that was just logged.

OWNER

The ID of the user who owns this test stream.

CREATED BY FUNCTION

LOG always appears in this field for a test stream that has just been logged.

TOTAL SCREENS IN: OUT:

The number of input and output screens.

AVERAGE BYTES IN: OUT:

The average number of bytes in the input and output data streams.

Exit Log Termination

Use the End (PF3) or Return (PF4) command to exit from the Log Termination menu.

End redisplays the Primary Options Menu, unless you entered Log Termination from the Inquiry function. In that case, CA Verify for CICS redisplays the Inquiry menu.

Return redisplays the Primary Options Menu.

Chapter 5: Run Function

This section contains the following topics:

Overview

The Run function lets you re-execute an application with the same data that was captured during logging.

CA Verify for CICS acts like a terminal operator, sending the stored input screens to the application, which responds with output. CA Verify for CICS then compares this output with the output from the logged test stream and flags any differences.

Use the Run function to:

- Test changes in a program
  - If the run completes normally, you know that the program is still processing the input data correctly.
  - If CA Verify for CICS interrupts the run because it detects differences between the logged screens and the current run screens, you can determine whether those differences are expected (because the program changed) or unexpected (because of an error).

    Note: Most anticipated changes can be identified using the Rules function; see the "Rules Function (see page 151)" chapter for details.
- Test changes in hardware or system software; for example, a new disk drive or a new CICS release
- Stress test your system; for example, run a large multiple terminal test stream to see how response time is affected.

How the Run Function Works

Here's how CA Verify for CICS runs a test stream:

- CA Verify for CICS allocates virtual terminals for all the terminals in the test stream you select, using the logged terminal status information to synchronize the status of the virtual terminal with that of the real terminal used in logging.
  - If the applications being run in a test stream require UserID security, the virtual terminal needs to go through signon processing. See the "Security" chapter in the *Installation Guide* for details on using automated signon for virtual terminals.
- CA Verify for CICS allocates an original display area, a modified display area, and a current display area for each selected terminal. These display areas represent the original terminal display as it appeared during logging, the original display areas as modified by any rules, and the current display as it would appear if the virtual terminal were real.
- CA Verify for CICS then processes each screen in the test stream as follows:
  - For each input screen, CA Verify for CICS enters the data and sends the data to the application. The input screen updates both the original and current display areas.
  - For each input screen that is affected by a rule, CA Verify for CICS creates an original expected screen showing the effect of the application of the rule.
  - For each output screen, CA Verify for CICS updates the original display area and waits for the corresponding output from the application. When the virtual terminal receives the output, CA Verify for CICS updates the current display area. CA Verify for CICS then compares the two output display areas.

If CA Verify for CICS detects a mismatch, it can:

- *Interrupt* the run. CA Verify for CICS identifies the differences between the expected and current screens. You can then display all three screens (original, expected, and current), correct the differences, terminate, or continue the run.

  Note: During a run, modeling can only be done on the record that has the mismatch.
- *Continue* the run. You can elect to save the original, current, and expected screens so the mismatches can be viewed when you subsequently browse or print the test stream.

You determine how CA Verify for CICS will execute the run. For example, you can specify:

- What to do with the output from the run; for example: discard it or save it as a new test stream
- What type of comparison CA Verify for CICS should perform; for example: physical, logical, or screen

- What to do with anticipated changes by designing a ruleset to be used when this test stream is run
- Whether or not the operator should explain a mismatch
- The rate at which input is sent to the application, simulating operator think time

Restore Test Data

When you run a test stream, you must use the same test data that was used when you logged the test stream. For example, if the logged test stream adds a customer record, the run test stream must be able to add the same record. Sometimes this means you will have to restore your test data.

- If the transaction you are testing does *not* modify the test data, you can use the original data. For example, you should be able to retest an inquiry transaction using the same unrestored data.
- If the transaction you are testing *does* modify the test data, you can either:
  - Back up the data before testing so it can be restored
  - Construct the test stream so that all changes are reversed at the end

    For example, delete any items added, add any items deleted, and reverse all updates.

Virtual Terminals

When CA Verify for CICS runs a test stream, it allocates virtual terminals for each terminal in the test stream. It uses terminal information, stored when the test stream was logged, to synchronize the status of the virtual terminal with the real terminal used in logging. Using virtual terminals:

- Saves system resources
- Does not restrict testing to the number of physical terminals available
- Does not affect users who need the real terminals

However, if you are using VTAM virtual terminals, you can specify that test streams be run with real terminals instead. Using real terminals may be useful when you are performing large-volume controlled tests. See Select a Terminal (see page 74) later in this chapter for instructions on running with real terminals.

Convert Variables to Rules

CA Verify for CICS Release 4.3 provides three types of variables: global, auto, and local. These variables are stored with each test stream. Starting with Release 4.4, these three types of variables have been incorporated into the Rules function. Variables, as well as all other types of field changes (like move field and change field value), can be defined using recognition and screen rules. See the "Rules Function (see page 151)" chapter for details on defining rulesets and the rules they contain.

Conversion Details

The first time a Release 4.3 test stream runs with the current release, the test stream will be reviewed to see if it has existing global, auto, or local variables. If it does, these variables will be automatically converted to the current release's rules that will be applied during the run.

Variables are converted as follows:

- *Auto* variables translate into a rule with Field Recognition criteria
- *Local* variables translate into a rule with Screen Recognition criteria
- *Global* variables translate into a rule with no recognition criteria (neither Field or Screen)

The resulting test stream will no longer contain any global, auto, or local variables, but will have a ruleset with the same name as the test stream associated with it.

In addition, if a Release 4.3 test stream is *copied* to a new test stream using the online utilities or a batch copy using the current release, the global, auto, and local variables will be converted into a ruleset with a matching name—just as if the test stream had been run using the current release. This is the recommended method if you have Release 4.3 test streams that will need to be manipulated using the Rules function.

Note: If a Release 4.3 test stream has run under the current release, it can still run in a Release 4.3 environment; however, any test stream created with a newer release *cannot* run in Release 4.3.

Invoke the Run Function

To run a test stream, type **R** on the Primary Options Menu and press Enter. CA Verify for CICS the Run Options menu.

The default ddname, Application, and Version values appear; if necessary, these can be changed.

```
------------------------------  RUN OPTIONS  -------------------------12:54:57
ENTER COMMAND ===>

ENTER INPUT TEST STREAM NAME:
  DDNAME      ===> TCADS
  APPLICATION ===>                  (LEAVE APPLICATION, MEMBER,
  MEMBER      ===>                   OR VERSION BLANK AND PRESS
  VERSION     ===> 001               ENTER FOR A SELECTION LIST)

 CREATE NEW OUTPUT TEST STREAM:
  DDNAME      ===>                  ("*" USES NEXT AVAILABLE VERSION)
  APPLICATION ===>
  MEMBER      ===>
  VERSION     ===>

PROCESS WITH RULES          ===> Y       (Y/N OR S-SELECT)
COMPARISON TYPE             ===> S       (S-SCREEN, L-LOGICAL, P-PHYSICAL)
RECORD HISTORY              ===> Y       (Y/N)
  REQUIRE SIGNOFF DATA      ===> N       (Y/N)
SIMULATED USER THINK TIME   ===> NONE    (NONE, NNN% OF ORIGINAL, NN SECONDS)
STATUS INTERVAL             ===> 005     (SECONDS)
CANCEL INTERVAL             ===> 001     (MINUTES)
STOP AT MISMATCHES          ===> Y       (Y/N)

F1-HELP        F3-END       F4-RETURN
```

Select a Test Stream

If you know which test stream you want to run, follow these steps:

1. Key in the Application name, Member name and other identifiers, if necessary.
2. Indicate whether a ruleset is to be used when this test stream is run.
3. Specify the options on this menu.
4. Press Enter.

   CA Verify for CICS runs the test stream and displays the Run Status menu.

If you don't know which test stream you want to run, follow these steps:

1. Leave the Application and Member field blank.
2. Optionally, blank out other identifiers.
3. Press Enter.

   CA Verify for CICS displays the Test Stream Selection menu.

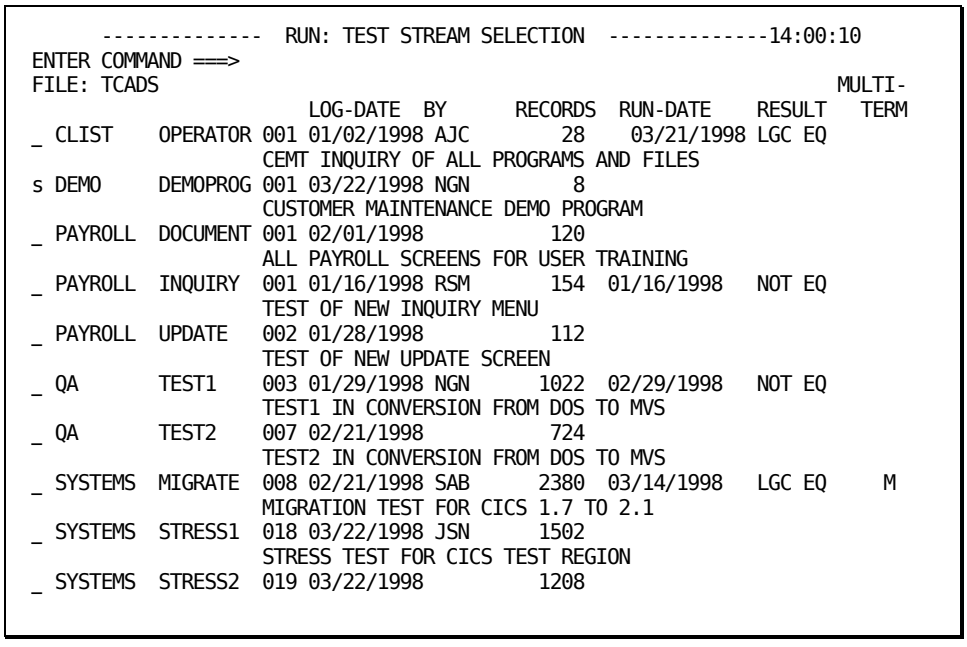
Select a Test Stream From the Test Stream Selection Menu

The Test Stream Selection menu lists all the test streams for the specified ddname.

Your position in the list depends on your specifications on the Run Options menu.

- If both the Application and Member fields are blank, CA Verify for CICS positions you at the first test stream for the specified ddname.
- If you leave only the Member field blank, CA Verify for CICS positions you just before the first test stream for the specified ddname and Application.
- If you key in a partial name, CA Verify for CICS positions you accordingly. For example, if you specify PAY as the Member name and blank out the Version number, CA Verify for CICS positions you just before the first test stream whose member name begins with 'PAY'.

A sample Test Stream Selection list is illustrated in the following example:



```
     --------------  RUN: TEST STREAM SELECTION  --------------14:00:10
ENTER COMMAND ===>
FILE: TCADS                                                          MULTI-
                        LOG-DATE  BY      RECORDS  RUN-DATE    RESULT   TERM
_ CLIST    OPERATOR 001 01/02/1998 AJC        28    03/21/1998 LGC EQ
                    CEMT INQUIRY OF ALL PROGRAMS AND FILES
s DEMO     DEMOPROG 001 03/22/1998 NGN         8
                    CUSTOMER MAINTENANCE DEMO PROGRAM
_ PAYROLL  DOCUMENT 001 02/01/1998           120
                    ALL PAYROLL SCREENS FOR USER TRAINING
_ PAYROLL  INQUIRY  001 01/16/1998 RSM       154  01/16/1998   NOT EQ
                    TEST OF NEW INQUIRY MENU
_ PAYROLL  UPDATE   002 01/28/1998           112
                    TEST OF NEW UPDATE SCREEN
_ QA       TEST1    003 01/29/1998 NGN      1022  02/29/1998   NOT EQ
                    TEST1 IN CONVERSION FROM DOS TO MVS
_ QA       TEST2    007 02/21/1998           724
                    TEST2 IN CONVERSION FROM DOS TO MVS
_ SYSTEMS  MIGRATE  008 02/21/1998 SAB      2380  03/14/1998   LGC EQ     M
                    MIGRATION TEST FOR CICS 1.7 TO 2.1
_ SYSTEMS  STRESS1  018 03/22/1998 JSN      1502
                    STRESS TEST FOR CICS TEST REGION
_ SYSTEMS  STRESS2  019 03/22/1998          1208
```

Type **S** to the left of the test stream you want to run and press Enter. The Run Options menu is redisplayed with the name of the test stream you selected.

The Test Stream Selection menu provides the following information for each test stream:

- Its name: Application, Member, and Version
- The date on which it was logged
- The ID of the operator who initiated the log
- The number of records in the test stream
- The result of the last run:

  LGC EQ

  There was a mismatch, but you did not stop at it because of variables or rules which automatically resolved the mismatch

  PHY EQ

  The test streams are byte-for-byte physically equal

  NOT EQ

  There was a mismatch during the run, which you stopped at
- The description of the test stream specified when it was logged
- **M** to indicate a multiple terminal test stream.

CREATE NEW OUTPUT TEST STREAM

This option controls the disposition of the run output. The following chart illustrates how these options should be specified.

Create New Output	Disposition of Run Output
Blank	Run output test stream and the output ruleset is discarded.
Specified	Run output creates new output test stream and new output ruleset (if rulesets are being used); input test stream and ruleset remain unchanged.

The default is blank (to discard the output test stream and output ruleset). To use this option you must also set the Record History option to "N" (because History is stored in the output test stream).

Follow these steps:

1. Specify the entire name or let CA Verify for CICS generate it as follows:
   - Type an asterisk (*) as the first character of the ddname to generate the same test stream name with the Version incremented by one.
   - Type an equal sign (=) for any part of the name — for example, ddname, Application, Member, or Version — to instruct CA Verify for CICS to use that part of the name from the input test stream. For example, if you specify = for Member, CA Verify for CICS will use the input Member name for the new output Member name.
2. Press Enter after specifying the new output name.

   CA Verify for CICS generates the name and redisplays the Run Options menu with the new name.
   - When an input test stream contains only input screens and an output test stream is created, the output test stream will contain both input and output screens.
   - When a new output test stream is created, a new output ruleset is also created (if rulesets are being used). This ruleset will match the output test stream and will contain just the recognition criteria and the variables from the input ruleset.

Specify Run Options

Most of the options on the Run Options menu contain default values selected during installation. Change these options to meet your testing needs.

PROCESS WITH RULES

Rulesets can be defined and associated with test streams to eliminate unnecessary RUN MISMATCHES of expected differences. See the "Rules Function (see page 151)" chapter for details on how to create and maintain rulesets and rules.

Y

Runs the test stream with the matching test stream ruleset and/or system ruleset and/or application ruleset. This is the default.

Note: The test stream ruleset must have the same name as the test stream.

N

Runs the test stream without applying any rulesets.

S

Displays the Ruleset Run Selection panel prior to initiating the run, where you can review and select the rulesets associated with the selected input test stream. See the section Select Rulesets for a Run *(see page 73)* for more information.

COMPARISON TYPE

Identifies the type of comparison CA Verify for CICS will perform:

S (Screen)

A row-by-row comparison of the final screen image as presented to you, regardless of the number of I/Os. This is the recommended comparison type for most situations.

Use S, the default, to run test streams associated with rules. This type of comparison is similar to a logical comparison which compares corresponding rows of the two screens to see it they're the same. However, the comparison base includes all output screens between two input screens.

L (Logical)

A row-by-row comparison of the screen image for each I/O.

A logical comparison compares the corresponding rows on the two screens to see if they're the same. This type of comparison allows variation in the bytes being transmitted as long as the *end result* is the same.

P (Physical)

A byte-for-byte comparison of the data stream.

Use physical to compare the data streams to determine if identical screen images were produced in the same way, for non-3270 terminals, or for test streams that use graphics. In a physical comparison, 80-byte segments of the data streams are compared.

RECORD HISTORY

Determines the contents of the output test stream after the run.

Y

The output test stream will contain the screens that will be used the next time it is run or edited, as well as historical information. Record History can include: original screens, expected screens, rules, differences, and signoff data. If you specify Y, you must create a new output test stream.

N

The output test stream will contain only the screens that will be used the next time it is run or edited.

REQUIRE SIGNOFF DATA

Y

Requires the operator to explain each mismatch when CA Verify for CICS compares the original, expected, and current screens. For example, you may want to require an explanation during a final Quality Assurance run. This option is not valid if Record History is N (no).

N

No explanation of mismatches is required. This is the default.

Note: Only functional if Stop At Mismatches = Y

SIMULATED USER THINK TIME

Controls the rate at which CA Verify for CICS sends input screens to the application, simulating your think time

None

Specifies the minimum think time. CA Verify for CICS will not wait before sending screens to the application. This is the default.

nnn%

Specifies the think time as a percentage of the original user think time. For example, 50% specifies half the original think time, 100% specifies the same think time, and 200% specifies twice the original think time. 200 is the maximum percentage that can be specified.

Nn

Specifies the think time in seconds. CA Verify for CICS will wait the same number of seconds before sending each input screen. 20 seconds is the maximum specification allowed.

STATUS INTERVAL

Determines the number of seconds between updates in the Run Status menu. While the test stream is running, CA Verify for CICS periodically updates this menu to indicate the number of records processed and the estimated remaining run time. The default is five seconds.

CANCEL INTERVAL

Determines the number of minutes between opportunities to cancel the run. During a long test stream run, CA Verify for CICS periodically pauses, issues a terminal read, and displays a message on the Run Status menu which allows you to cancel the run. Run processing is suspended while you react to this message. The default is one minute.

You can also use the Inquiry function to interrupt a run which will not issue a terminal read for several minutes. See the "Inquiry Function (see page 205)" chapter for details.

STOP AT MISMATCHES

Y

When Y is selected the run will stop if any mismatches are encountered. The default is Y.

N

When N is selected the run will not stop at mismatches.

Select Rulesets for a Run

Follow these steps:

1. Specify the run option for Process With Rules as S (SELECT).\
2. Press Enter.

   CA Verify for CICS displays the Ruleset Run Selection panel, as shown following.

   Use this panel to review or choose which rulesets to use for this run.

```
---------------------------RULESET RUN SELECTION----------------------08:20:58
 ENTER COMMAND ===>

 USE

 Y TEST STREAM RULESET: TCADS.CCCORDER.REL63.001
     DESCRIPTION: CAROL'S COOKIES COMPANY
                  CONVERSION FROM REL 6.3 TO 7.0


 N APPLICATION RULESET: NOT USED
     DESCRIPTION:



 Y SYSTEM RULESET:      TCADS.SYSTEM.#RULESET.001
     DESCRIPTION: GLOBAL RULES FOR ALL SCREENS



  F1-HELP  F3-END   F4-RETURN
```

TEST STREAM RULESET

Contains the name of the test stream ruleset associated with this test stream. Type Y to the left of the heading if this ruleset is to be used during this run; type N if it is not to be used.

APPLICATION RULESET

Contains the name of the application ruleset associated with this test stream. An application ruleset contains rules that apply to every test stream on the TCADS file that has the same Application name. Type Y to the left of the heading if this ruleset is to be used during this run; type N if it is not to be used.

Note: If there is no application ruleset, then NOT USED appears in this field.

SYSTEM RULESET

Contains the name of the system ruleset associated with this TCADS. A system ruleset contains rules that apply to every test stream on a specific TCADS file. Type Y to the left of the heading if this ruleset is to be used during this run; type N if it is not to be used.

Note: If there is no system ruleset, then NOT USED appears in this field.

Note: Rules will be applied in the order they are specified in the ruleset you select. If you select multiple rulesets, rules in the test stream ruleset will be applied first; rules in the application ruleset will be applied second; rules in the system ruleset will be applied last.

3. To initiate the Run with the Specified Rulesets named on this panel, press Enter. Continue with the instructions in the section Initiating the Run.
4. To exit back to the Run Options menu without processing the changes, press PF3 (End).

Initiate the Run

When you have finished specifying the run options, press Enter.

- If you are running a single terminal test stream, CA Verify for CICS initiates the run.
- If you are running a multiple terminal test stream, CA Verify for CICS displays the Terminal Selection menu.

Select a Terminal

CA Verify for CICS displays the Terminal Selection menu before you run a multiple terminal test stream.

```
SAMPLE.MULTTERM.001 -------  RUN: TERMINAL SELECTION  -----------------14:14:22
ENTER COMMAND ===>                                                          RT

SELECT:  "S"    ORIGINAL   TERMINAL      SCREEN      ALTERNATE   VIRTUAL
INQUIRY: "I"    TERMINAL   TYPE          SIZE        SIZE        TERMINAL

          _     A60L2048   3277 REMOTE   24 BY  80   24 BY  80
          _     A60L2049   3277 REMOTE   24 BY  80   24 BY  80
          _     A60L205A   3277 REMOTE   24 BY  80   24 BY  80
          _     A60L205B   3277 REMOTE   24 BY  80   24 BY  80
         ***  END OF TERMINALS  ***









TYPE AN "S" TO SELECT TERMINAL FOR RUN      TYPE AN "I" TO VIEW TERMINAL STATUS
F1-HELP        F3-CONTINUE   F4-RETURN      F7-UP          F8-DOWN
```

For each terminal in the test stream, CA Verify for CICS identifies its name, type, screen size, alternate size, and, if specified, the virtual terminal assignment.

You can run one, all, or a combination of terminals. You must select at least one terminal.

- Type **S** to the left of each terminal you want included in the run and specify Continue (PF3) to initiate the run.
- Type **I** and press Enter to view the Initial Terminal Status menu for a terminal. See the example in the "Browse Function (see page 111)" chapter for an explanation of this menu.

When you have finished viewing the Initial Terminal Status menu, use End (PF3) to redisplay the Terminal Selection menu. Then press PF3, Continue, to initiate the run.

Use the following CA Verify for CICS commands:

Assign

Inserts **S** to the left of each terminal

Reset

Restores the initial menu setting with an underscore to the left of each terminal

Cancel

Redisplays the Test Stream Selection list or Run Options menu

Up (PF7)

Scrolls through the terminal list

Down (PF8)

Scrolls through the terminal list

Run with Real Terminals

When using VTAM virtual terminals, CA Verify for CICS allows the specification of the type of terminals the test streams are to use during a run. The following commands must be issued from a clear screen and will stay in effect for any test stream run from a terminal issuing the command, or until a different XTCA TERMINALS command is issued.

XTCA TERMINALS VIRTUAL

Runs will execute only on virtual terminals defined in the CICS region. This is the default.

XTCA TERMINALS REAL

CA Verify for CICS will attempt to run on the original terminals the test stream was logged from. If original terminals are not available, that is, in use or not defined in the CICS region, CA Verify for CICS will use the next available real terminal entry. If there are not real terminal entries available or defined in the CICS region, CA Verify for CICS will use any available virtual terminal entries.

XTCA TERMINALS ORIGINAL

CA Verify for CICS will attempt to run on the original terminals the test stream was logged from. If the original terminals are unavailable, the test stream will not be run and a message will be displayed on the Run menu.

Review the Run Status

During the run, CA Verify for CICS displays the Run Status panel at the terminal where the Run function was initiated. CA Verify for CICS updates this panel at the status interval specified on the Run Options menu.

```
 ------------------------------  RUN STATUS  ----------------11:40:22
                                                                R3

 INPUT TEST STREAM:     TCADS.CCC.ORDERAPP.001

 OUTPUT TEST STREAM:    NONE

 RULESET(S):
   TESTSTREAM:  TCADS.CCCORDER.REL63.001
   APPLICATION: NOT USED
   SYSTEM:      TCADS.SYSTEM.#RULESET.001

 CURRENTLY PROCESSING RECORD:         1
 TOTAL RECORDS TO BE PROCESSED:       8
 ESTIMATED REMAINING RUN TIME: 00:00:36

 STATUS INTERVAL:  5 SECONDS
 CANCEL INTERVAL:  1 MINUTE

 DESCRIPTION: CAROL'S COOKIES COMPANY DEMO PROGRAM


F1-HELP     F3-END     ENTER-CONTINUE
```

This menu provides the following information:

INPUT TEST STREAM

Identifies the ddname, application, member, and version of the logged test stream, separated by periods.

OUTPUT TEST STREAM

Identifies the ddname, application, member, and version of the output test stream, separated by periods. None will appear if you did not specify a new name for the output test stream.

RULESET(S):

Identifies the rulesets CA Verify for CICS will apply to the input test stream during the current run processing

TESTSTREAM:

Applies changes only to this test stream

APPLICATION:

Applies changes to every test stream on the TCADS file that contains the same application name

SYSTEM:

Applies changes to every test stream on a specific TCADS file.

CURRENTLY PROCESSING RECORD

Identifies the record CA Verify for CICS is currently processing.

TOTAL RECORDS TO BE PROCESSED

Identifies the total number of records in the test stream.

ESTIMATED REMAINING RUN TIME

Estimates the amount of time required to complete the run, based on the total response time of the remaining records. Remember, this is only an estimate and may be inaccurate for many reasons; for example, for a multiple terminal test stream in which some terminals have been excluded.

STATUS INTERVAL

Indicates the number of seconds between updates of this menu.

CANCEL INTERVAL

Indicates the number of minutes between opportunities to cancel the run. When the cancel interval expires, a message at the bottom of the menu indicates that you can cancel or continue the run. Run processing is suspended while you react to this message.

DESCRIPTION

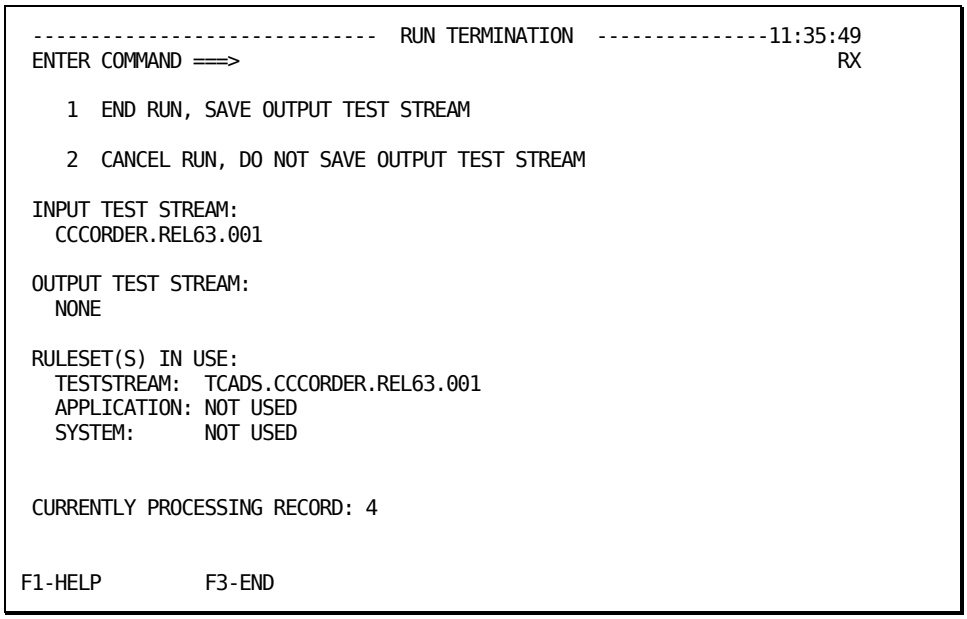
The description of the test stream specified when it was logged.

To continue the run, press Enter; to cancel the run, press any PF key or type End.

Note: The command line appears on the menu only when CA Verify for CICS displays the message allowing you to cancel the run.

Terminate the Run

If you cancel the run from the Run Status panel, CA Verify for CICS displays the Run Termination menu. You can also access this menu by using the End (PF3) command from the Run Mismatch Options menu, and the End (PF3) command from the Missing Output menus, discussed later in this chapter.



- Type **1** and press Enter to end the run and save the output test stream and ruleset.
- Type **2** and press Enter to cancel the run *without* saving the output test stream.
- Use the End (PF3) command to continue the run or return to the previous menu.

Specify Mismatch Options

When CA Verify for CICS detects a mismatch during a logical, physical, or screen comparison, it stops the run and displays the Run Mismatch Options menu. The mismatch can be any of the following:

- I/O difference; for example, a write instead of a read
- type of operation; for example, an erase/write instead of an erase/write alternate
- cursor location

- a byte of data or an attribute
- write control character

```
  --------------------------  RUN MISMATCH OPTIONS  --------------16:08:57
  ENTER COMMAND ===>                                                  R6

 1   1  DISPLAY RULESET SUMMARY             6    ACCEPT ORIGINAL OUTPUT
     2  DISPLAY PREVIOUS INPUT               7    ACCEPT CURRENT OUTPUT
     3  DISPLAY NEXT INPUT                   8    CHANGE NEXT INPUT
     4  SKIP ORIGINAL OUTPUT                 9    INSERT CURRENT OUTPUT AND
INPUT
     5  SKIP ORIGINAL OUTPUT AND INPUT       10   INSERT CURRENT OUTPUT

INPUT:   CCCORDER.REL63.001              CURRENT RECORD:  4
OUTPUT:  NONE
  2
RULESET: TESTSTREAM:  TCADS.CCCORDER.REL63.001
  3      APPLICATION: NOT USED
         SYSTEM:      NOT USED

          TYPE     OPERATION        WCC   CURSOR   SIZE    LENGTH   TERMINAL
EXPECTED: OUTPUT   ERASE/WRITE      C3     9   2   24*80      602   NN01
CURRENT:  OUTPUT   ERASE/WRITE      C3     9   2   24*80      602   VTERM001
ONLY UNEQUAL ROW: 02
                                .ORDER STATUS                .DATE.07/22/97
  4                             .ORDER STATUS                .DATE.09/11/1997
                                                             XXXXXXXXXXXXXX
  F1-HELP     F2-ROTATE     F3-END     F7-UP     F8-DOWN
```

The previous menu consists of four sections:

1. Lists the available options.
2. Describes the run.
3. Contains the rulesets used for this run.
4. Identifies the differences between the original/expected, and current screens that generated the mismatch.

Mismatch Options

To specify an option, type its number (or the corresponding command) and press Enter, or press the appropriate PF key. The options consist of the following groups:

Option 1:

Displays the rules summary menu for viewing or updating the rules being applied to the input test stream.

Options 2-3:

Displays the previous or next input

Options 4-5:

Skips the original output screen, or the next output/input screen pair

Options 6-7:

Ignore or accept the change; if you accept the change, the output test stream is updated for use in the next edit or run

Option 8:

Displays the screen for you to update

Option 9-10:

Adds a new output/input screen pair, or a new output screen

When you use options 2-3 to display screens, you can use the Format (PF9) command to view the screens in Display, Field, and Hex formats. See the Display, Field, and Hex Screen Formats (see page 97) section later in this chapter for a discussion of screen formats.

To scroll forward through a test stream, use the Next (PF6) command to display the next screen. However, you cannot scroll backward. If you need to see earlier screens, suspend the Run function and use the Browse function to inspect them.

The mismatch options are described next.

1 DISPLAY RULESET SUMMARY

Instructs CA Verify for CICS to display a summary of the rulesets applied during the run. This option allows you to define a new rule or update an existing rule and save your updates to the current ruleset.

If you modify the existing ruleset and specify End (PF3), CA Verify for CICS will repeat the comparison, taking into account your rule specification. If CA Verify for CICS no longer detects a mismatch, it will continue the run; otherwise, it will redisplay the Run Mismatch Options menu. For example, if the run was interrupted because of a discrepancy between the dates on the original and current screens and you identify the date field as a variable field, CA Verify for CICS will continue the run

See the "Rules Function (see page 151)" chapter for instructions on how to define a ruleset and the rules it contains; see the section Adjust Rules During a Run later in this chapter for instructions on how to change or add rules to a ruleset during a run.

2 DISPLAY PREVIOUS INPUT

Instructs CA Verify for CICS to display the last screen sent to the application. Specify End (PF3) to return to the Run Mismatch Options menu.

3 DISPLAY NEXT INPUT

Instructs CA Verify for CICS to display the next screen to be sent to the application. Specify Next (PF6) to display subsequent screens in the input test stream; specify End (PF3) to return to the Run Mismatch Options menu.

4 SKIP ORIGINAL OUTPUT

Use this option when the input test stream has two consecutive output screens but one has been deleted by the application. This option deletes the first output screen so it will not be used the next time the output test stream is run, browsed, or edited. CA Verify for CICS then repeats the comparison, using the next screen in the original test stream and the current screen. If there is no mismatch, the run continues; otherwise, CA Verify for CICS redisplays the Run Mismatch Options menu.

In the previous example, if record 4 was in the test stream but had been eliminated from the application, you would use option 4 to delete it.

5 SKIP ORIGINAL OUTPUT AND INPUT

Use this option when an output and input screen have been deleted from the application. This option deletes the output screen and any input screens up to the next output screen so they will not be used the next time the output test stream is run, browsed, or edited. CA Verify for CICS then repeats the comparison, using the next output screen in the input test stream and the current screen. If there is no mismatch, the run continues; otherwise, CA Verify for CICS redisplays the Run Mismatch Options menu.

In the previous example, if records 2 and 3 were in the test stream but had been eliminated from the application, you would use option 5 to delete them.

6 ACCEPT ORIGINAL OUTPUT

Instructs CA Verify for CICS to retain the original screen in the output test stream and continue the run. Use this option when the output screen generated by the application reflects an error and the original screen is correct.

7 ACCEPT CURRENT OUTPUT

Instructs CA Verify for CICS to replace the original screen with the current screen in the output test stream and continue the run. Use this option when the application has changed and the original screen is no longer correct.

8 CHANGE NEXT INPUT

Select this option when you want to change the next *input* screen required by the application. This may be necessary when the current output screen changes the location or contents of a field which would be part of the next input screen, or when an input field has been added to the screen. Option 8 ensures that the application receives the data it expects.

For example, suppose your next input screen should select menu item C. However, your application has been updated and what was formerly item C is now item D. Option 8 lets you replace the original input screen with an updated input screen which specifies item D.

When you select option 8, CA Verify for CICS displays the current screen. Type the necessary data and press the appropriate key (attention identifier) required by the application (like Enter or PF1). Whatever key you use will be interpreted according to the application specifications. CA Verify for CICS sends the input to the application and will use this data the next time you run or edit the output test stream.

When you choose option 8, CA Verify for CICS accepts the mismatch on the current screen and continues the run.

Note: Before specifying option 8, you may need to select option 3 to display the next input screen so you know what input the application expects. Use the Format (PF9) command to display the screen in Field format, which shows only the input fields and identifies the AID (attention identifier key). Then return to the Run Mismatch Options menu, select option 8, and type the required input data.

See the section Use Options 8 and 9 With Non-3270 Devices for additional information.

9 INSERT CURRENT OUTPUT AND INPUT

Use this option to insert an output screen and an input screen. CA Verify for CICS displays the current output screen. Type the input data and press the appropriate key as required by the application. CA Verify for CICS will insert the new screen into the test stream and send it to the application. CA Verify for CICS also holds the original screen because CA Verify for CICS assumes it will match the next output screen from the application.

In the previous example, if records 2 and 3 were not in the test stream but had been added to the application, you would use option 9 to insert them.

See the section Use Options 8 and 9 With Non-3270 Devices (see page 83) for additional information.

10 INSERT CURRENT OUTPUT

Use this option to insert the current screen into the output test stream. CA Verify for CICS then repeats the comparison, using the next screen from the application. You can choose this option only if the application has already sent a second output screen.

In the previous example, if record 4 was not in the test stream but had been added to the application, you would use this option to add it.

Note: When new or deleted screens are anticipated, use the Rules function to identify the screens to be added or removed. The Rules function is discussed in the "Rules Function (see page 151)" chapter.

Use Options 8 and 9 with Non-3270 Devices

Options 8 and 9 work differently with non-3270 devices:

- CA Verify for CICS does not display the current screen, so you should use PF2 to rotate to the current screen to examine it *before* invoking option 8 or 9.
- Non-3270 data streams often contain hexadecimal control information which varies, depending on the devices and controller programs used. You must enter this information with the character data. CA Verify for CICS will expect:
  `X'hex data',C'character data',X'hex data' . . .`

  CA Verify for CICS will reject data which does not conform to this format and ask for the input again. To determine the format of non-3270 data at your installation, you can request a batch print of a data stream using the Hex option or you can browse it in hex format.

Description of the Run

The information following the options on the Run Mismatch Options menu describes the run.

INPUT

Identifies the input test stream being run.

OUTPUT

Identifies the output test stream being created; NONE if you are not saving the output from the run.

CURRENT RECORD

Identifies the record number of the screen in the input test stream. If your test stream consists of an input screen followed by an output screen, all input screens will be odd-numbered and all output screens will be even-numbered.

Description of the Rulesets

The information following the run description on the Run Mismatch Options menu identifies the ruleset(s) being used when this test stream is run.

TESTSTREAM

The name of the test stream ruleset, if one is used.

APPLICATION

The name of the application ruleset, if one is used.

SYSTEM

The name of the system ruleset, if one is used.

Important! To invoke rulesets during a log or a run RULESYS=YES and RULEAPP=YES must be set in the TCAOPTS Macro Options. For more information see the *Installation Guide*.

Description of the Screens

The information following Ruleset on the Run Mismatch Options menu describes the original, expected, and current screens where the mismatch occurs.

Type

Input or output. Usually, both will be output screens.

Operation

Type of 3270 operation; for example, write, erase/write. A logical or screen comparison does not compare the operation because any difference will be detected during the row comparison.

WCC

Two-character hex representation of the write control character. If the two WCCs differ, their meanings are listed beneath the First Unequal Row information.

Cursor

Location of the cursor by row and column. Cursor location is included in logical and screen comparisons; in a physical comparison the difference will be detected in the data stream comparison.

Size

Screen size used when the screen was received from the application. Both logical and physical comparisons compare screen size. If screens of different sizes are used, the logical comparison pads the rows and columns of the smaller screen with nulls.

Length

Number of bytes in the data stream. A physical comparison compares data length.

Terminal

Name of the terminal that received the screen from the application. CA Verify for CICS does not compare the terminal, but this information is provided when you run a multiple terminal test stream so that you know which terminal the screens are for.

Unequal Row Information

The unequal row information at the bottom of the Run Mismatch Options menu describes the first or only mismatch between the original/expected and current screens.

- During a *logical* or *screen* comparison, CA Verify for CICS compares the rows. When it detects a mismatch, it displays unequal row information.
- During a *physical* comparison, CA Verify for CICS compares the two data streams without formatting them, and consequently compares 80-byte segments rather than rows. When CA Verify for CICS detects a mismatch, it displays unequal segment information. If the two data streams are unequal in length, CA Verify for CICS pads the shorter stream with underscores.

CA Verify for CICS displays the unequal data when it detects a mismatch in the data or attributes. When it detects a different type of mismatch — for example, differences in the operation or write control character — it displays a message explaining the mismatch.

FIRST/ONLY UNEQUAL ROW (SEGMENT)

Identifies the row in which the first mismatch was detected. For a physical comparison, this is the offset of the first unequal segment, relative to zero.

UNEQUAL ROWS (SEGMENTS)

Identifies how many rows or segments are unequal.

OFFSET

For a physical comparison only, identifies the offset of the 80-byte segment displayed, relative to zero.

Data in Unequal Rows (Segments)

Beneath the *First/Only Unequal Row i*nformation, CA Verify for CICS displays three lines:

Line 1

Row or segment from the original or the expected original screen

Line 2

Row or segment from the current screen

Line 3

Differences between the rows or segments, specified as follows:

Blank

The two rows or segments contain the same characters

X

The two rows or segments contain different characters

-

Variable field; not compared

Note: In the original/expected and current rows, CA Verify for CICS substitutes a blank for nulls and a period for unprintable characters (usually attributes). An **X** beneath two periods means the attributes are different; display the screens using Field format to view the attributes. An **X** beneath two blanks means one is a null and the other is a blank.

If the screens have more than one set of unequal rows, use the Down (PF8) and Up (PF7) commands to scroll through all the additional unequal rows. If the width of the test stream screen exceeds the width of the terminal you are using, use the Left (PF10) and Right (PF11) commands to display the entire screen.

When you page down through the sets of unequal rows, the menu you are viewing is titled *All Unequal Rows*. This menu has a column ruler to help you determine the location of the mismatch. You can key in all mismatch options on this menu. Use the Rotate (PF2) command to rotate between the original, expected, and current screens and the All Unequal Rows menu.

If the following mismatches are found, CA Verify for CICS displays this information after the last unequal screen row:

WCC

Explains the difference between the WCC's; for example:

```
Expected:  NL/EM/CR Print Cntl, Alarm
Current:  NL/EM/CR Print Cntl, Resetmdt
The Write Control Character has changed
```

Screen Size

Explains the difference in screen size; for example:

```
Expected:  24 * 80
Current:  27 * 132
The screen size in use has changed
```

Cursor

Explains the difference in cursor position; for example:

```
Expected:  Row: 3 Column: 8
Current:  Row: 1 Column: 1
The cursor location has changed
```

Operation

Explains the difference in operation; for example:

```
Expected:  Erase/Write
Current:  Erase/Write Alternate
The operation has changed
```

Type of Screen

If CA Verify for CICS detects a difference in the type of screen (for example, the original screen is an input screen, the current screen is an output screen), it displays the following message:

```
The application has issued a write when a read was expected. Either the write has been
added or a read has been eliminated. Select option 6, 8, or 10.
```

This is an I/O mismatch, caused by a change in the application or the deletion of an output screen during editing. Only the options identified in the message can be used. Select option:

- **6** if you don't expect the write to be issued again
- **8** to accept the current output screen and change the input screen which follows this output screen
- **10** to accept the new output screen and continue

Note: When new or deleted screens are anticipated, use the Rules function to identify the screens to be added or removed. The Rules function is discussed in a later chapter.

Sign off on the Mismatch

If you specified Y for both the Record History and Require Signoff Data options on the Run Options menu, CA Verify for CICS provides three lines beneath each mismatch in which you can explain the difference. Press PF8 from a mismatch display to view the unequal rows.

The following example illustrates how you would specify mismatch signoff information.

```
 CCC.ORDERAPP.001 -----  RUN: ALL UNEQUAL ROWS  ---------------11:18:23
 ENTER COMMAND ===>                                                RL

                                  RECORD:     2
       10       20       30       40       50       60       70       80
ROW: 1---+--------+--------+--------+--------+--------+--------+--------+
.DATE.02/06/1998            CAROL'S COOKIES COMPANY         .RELEASE 7.0
.DATE.02/10/1998            CAROL'S COOKIES COMPANY         .RELEASE 6.2
             XX
SIGNOFF  ===> The difference in the date field will
ON ABOVE ===> be resolved by defining a variable field
         ===>                                              OPER ===> NGN
         ***  END OF UNEQUAL ROWS  ***









  F1-HELP     F2-ROTATE       F3-END   F7-UP    F8-DOWN  F10-LEFT F11-RIGHT
```

Notes:

- If you select mismatch option 6, 7, 8, or 9, you must type at least one non-blank character in the first Signoff On Above field. If there are additional mismatch rows *on the same screen,* you can specify an asterisk (*) in subsequent Signoff On Above fields to indicate that the initial information be copied. An asterisk cannot be used to copy signoff information from another screen.
- Your signon operator ID appears by default in the OPER field. If necessary, you can change this ID.

Adjust Rules During a Run

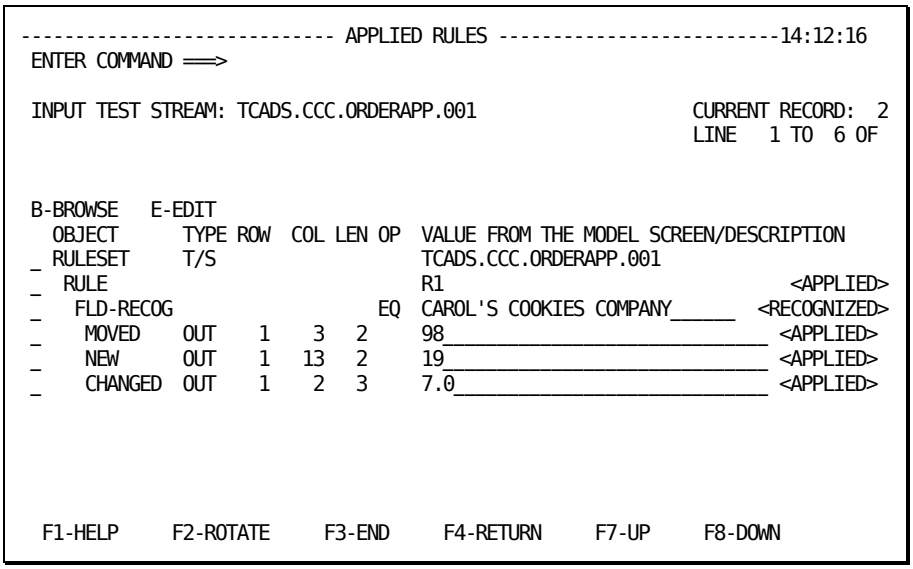
To help eliminate mismatches during a run, you may occasionally need to change an existing rule or add another rule to the ruleset for a test stream. (A common example is forgetting to create a rule for a change.)

Follow these steps:

1. Select option 1 (Display Ruleset Summary) from the Run Mismatch Options menu (shown on page 5-17).

   The Applied Rules panel is displayed as shown next.

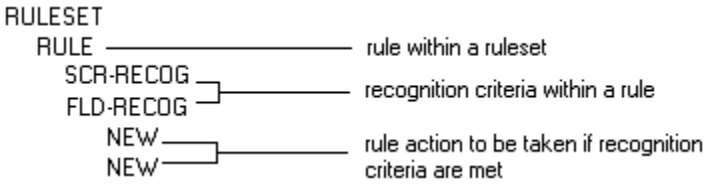
Note: When adding a rule to a ruleset, remember that rules are applied in the order specified in the ruleset.



This panel is similar to the Browse Rules—Summary panel, except that it lists all of the rulesets associated with this test stream and indicates which rule within those rulesets has been recognized or applied. Fields that are important for adjusting rules during a run are explained next; see the "Rules Function" chapter for an explanation of the rest of the fields.

OBJECT

This column lists the rulesets, rules, and rule actions in the order they are to be applied. Objects are arranged in hierarchical order with ruleset at the top of the hierarchy and rule actions being at the bottom. An illustration of the hierarchy appears next.



TYPE

Indicates which type of ruleset or rule this is. Valid types for ruleset are:

T/S

Test Stream

APPL

Application

SYST

System

OUT

Applies to this output screen

IN

Applies to the next input screen

<APPLIED>
<RECOGNIZED>

These messages indicate whether the rule has been applied during this run (<APPLIED>), or whether screens have been located that match the recognition criteria specified (<RECOGNIZED>).

2. To help identify any remaining changes, use PF2 (Rotate) to switch the display from the Expected screen to the Current view to the Original test stream record and then back to the Expected screen.
   - The *expected screen* contains the originally-logged test stream with the rules applied.
   - The *current view* shows the test stream record as it is now.
   - The *original test stream r*ecord displays the record from the originally-logged test stream with no changes applied.
3. If you have the proper security authorizations, you can take several actions from this panel. You can:
   - Browse or edit any ruleset by typing a B or E to its left. From there, browse or edit as needed.
   - Browse or edit any rule by typing a B or E to the left of RULE.
   - Browse or edit any rule action by typing a B or E to the left of the rule action.
   - Add a rule or rule action by typing an E to the left of the ruleset that will contain the rule. See the next section, Add a Rule or Rule Action, for details.

See the chapter "Rules Function" for detailed instructions on browsing, editing, and deleting rules and rule actions. The procedure for adding a rule or rule action is covered next.

Add a Rule or Rule Action

Follow these steps:

1. From the Applied Rules panel, type an **E** next to the ruleset that you want to add the rule or rule action to. Press Enter to display the Edit Rules—Summary panel.
2. Type an **I** (for Insert) to the left of the object that is to immediately precede the new rule or rule action. Press Enter to display the Rules Menu.
3. Add this rule or rule action as you would any other rule or rule action. (Detailed instructions are in Chapter 8, *Rules Function*.)

   Note: Rule actions are processed in the order in which they are defined, so it is important to insert a new item in the correct position.
4. After you type the specific information for this field (i.e., ROW, COL, LEN), press Enter. If the add was successful, the message ADD SUCCESSFUL appears in the message area.
5. Press PF3 to return to the Edit Rules—Summary where you can check the details of the rule or rule action you just added.
6. Press PF3 to display the Applied Rules to check whether the rule or rule action was applied. (Rules and rule actions are applied as soon as you add them.)
7. Continue with the run.

Resolve Conflicts between Rule Actions

Whenever CA Verify for CICS detects a conflict between rule actions, the run stops and the Mismatch screen is displayed. Using this screen, you must correct any conflicts between rule actions before the run can continue.

Rules Summary Status Messages

The following error messages may appear on the Rules Summary panel and the Rules Applied During Run panel after running a test stream.

<APPLIED>

Reason**:**

This rule or rule action has been applied.

<CONFLICT>

Reason:

One or more rule actions applied to the same location and a conflict occurred.

<ERROR>

Reason:

This is a general error code set when the reason code is unknown.

<INCOMPLETE RULE>

Reason:

The rule does not have any rule actions defined to it.

<INV DATE MASK-ERROR>

Reason:

The data in the specified data generation field does not match the mask specified in the date aging format field.

<INV NUMBER-ERROR>

Reason:

The data specified in the data generation field is not numeric.

<M,D,Y IS NOT NUM-ERROR>

Reason:

The month, day, or year in the data generation field location is not numeric.

<MULTI ACTION-ERROR>

Reason:

More than one delete screen or insert test stream rule action was specified for the same screen.

<NEGATIVE DATE-ERROR>

Reason:

The date generated has a negative value.

<NESTED ACTION-ERROR>

Reason:

A nested delete or insert test stream rule action was specified, which is not allowed.

<NO ATTR ROW-COL-ERROR>

Reason:

An attribute byte was missing for this row-column location.

<NO CUT DATA-ERROR>

Reason:

The requested cut data does not exist, so it cannot be pasted.

<NO MODEL SCREEN>

Reason:

No model screen applies to this rule.

<NOT APPLICABLE DURING RUN>

Reason:

This is a LOGGING rule action and is ignored during a run.

<NOT APPLIED>

Reason:

This rule or rule action has not been applied.

<NOT RECOGNIZED>

Reason:

The recognition criteria did not match.

<NOT RECOGNIZED-NOT APPLIED>

Reason:

The recognition criteria did not match, so the rule was not applied.

<NUM OVERFLOW-ERROR>

Reason:

An overflow condition occurs when the value of a field is too large to fit in the field length specified on the data generation rule action. Either re-specify the length or set OVERFLOW equal to YES.

<OUTBOUNDS-ERROR>

Reason:

The row, column, and length coordinates of this rule action are outside the screen size of the terminal used by Run.

<OUTBOUNDS-NOT RECOGNIZED>

Reason:

The row, column, and length coordinates of the recognition criteria are outside the screen size of the terminal used by Run.

<RECOGNIZED>

Reason:

The recognition criteria matched.

<SCREEN FAILURE-ERROR>

Reason:

A screen failure occurred because the attributes for the expected screen did not match those of the current screen. Check that all rules for deletes, moves, and changed fields are correct. The row, column, and length coordinates of this rule action overlap the position where the attribute mismatch was detected.

<SCREENID NOT EQ-NOT APPLIED>

Reason:

The specified screen ID did not match, so the rule was not applied

<T/S EMPTY-ERROR>

Reason:

The test stream to be inserted is empty.

<T/S IN USE-ERROR>

Reason:

The test stream to be inserted is in use.

<T/S I/O-ERROR>

Reason:

An I/O error occurred reading the test stream.

<T/S IS MTTS-ERROR>

Reason:

The test stream to be inserted is a multiple terminal test stream.

<T/S NOTAUTH-ERROR>

Reason:

The test stream to be inserted is READ protected.

<T/S NOTFOUND-ERROR>

Reason:

The test stream to be inserted is not in the TCADS

<T/S REC#1 NOT OUTPUT-ERROR>

Reason:

The first record in the test stream is not an output record.

VARIABLE CONFLICT - WARNING

Reason:

A variable rule action overlapped row and column location with a non-variable rule action; an informational warning.

Action:

Verify that the overlap is what you intended, then continue the Run. The conflict does not have to be resolved or bypassed.

WARNING

Reason:

An informational warning.

Action:

Verify that the overlap is what you intended, then continue the Run. The conflict does not have to be resolved or bypassed.

Errors Requiring CA Support

If any of the following errors occur, you must call CA support.

<INV OPER VALUE-ERROR>

Reason:

The data generation operator value is zero or not numeric.

<JULIAN DATE MASK-ERROR>

Reason:

The Julian date mask cannot contain a month.

<MULT M,D,Y IN MASK-ERROR>

Reason:

More than one MM, DD, or YY[YY] was specified in the date mask.

<ZERO OPER VALUE-ERROR>

Reason:

The operator value was zero.

Confirm the Mismatch

If you select options 4, 6, 7, 8, or 10 on the Run Mismatch Options menu, CA Verify for CICS validates the next input screen to check for the following conditions:

- the input data does not fit in the corresponding field in the current output screen
- the location of an input field has changed
- an input field was deleted

Any of these conditions will probably generate a mapping error (mapfail condition) when the next input screen is sent to the application. To avoid this, CA Verify for CICS displays the Mismatch Confirmation panel.

```
-----------------  RUN: MISMATCH CONFIRMATION  -----------------11:49:50
ENTER COMMAND ===>                                                    RC

                                                    RECORD NUMBER:     2
THE OPTION YOU SELECTED (6) MAY RESULT IN A MAPPING ERROR (MAPFAIL
CONDITION) WHEN THE NEXT INPUT IS SENT TO THE APPLICATION.

1 INPUT FIELD NO LONGER CORRESPONDS (BECAUSE 1 OR MORE FIELDS HAVE BEEN
MOVED, DELETED, OR SHORTENED). THE FIRST INPUT DATA WHICH DOES NOT
CORRESPOND TO ANY FIELD ON THE CURRENT SCREEN IS AT ROW 8, COLUMN 51.
IT IS 1 BYTE LONG.

MISMATCH OPTION 8 CAN BE USED TO ENTER THE NEXT INPUT INTO THE NEW SCREEN
LOCATIONS.

OPTION 5 OR 9 MAY ALSO BE APPLICABLE (IF SCREENS HAVE BEEN INSERTED OR
DELETED).



F1-HELP     F3-END
```

Usually, you can correct the error by selecting option 8 on the Run Mismatch Options menu. This option displays the current output screen so you can key in the input data and submit that screen in place of the original input screen. If the error occurred because screens were removed from the application, select option 5; if the error occurred because screens were added, select option 9.

Type the End (PF3) command to return to the Run Mismatch Options menu to select another option. Press Enter to continue the run, using the option you selected.

Display, Field, and Hex Screen Formats

When you select mismatch options 2 or 3, or use the Rotate command (PF2), CA Verify for CICS displays an input or output screen. You can view screens in three formats:

- Display
- Field
- Hex

Change Formats

Use the Format (PF9) command to shift between the Display, Field, and Hex formats. Although parameters are not required, you can specify Format xxx, replacing xxx with one of the following:

Display

Immediately shifts to Display format

Field

Immediately shifts to Field format

Hex

Immediately shifts to Hex format

NODisplay

Disables Display format

NOField

Disables Field format

NOHex

Disables Hex format

For example, Format F displays the screen in field format.

Note: If the format being disabled is the one currently in use, CA Verify for CICS automatically switches to the next available format. If the current format is the only one enabled, CA Verify for CICS displays an error message and does not alter the display.

Display Format

Display format is the default format for viewing a screen. A sample screen in Display format is illustrated next.

```
CCC.ORDERAPP.001 -----------  RUN: CURRENT SCREEN  --------------------13:37:07
ENTER COMMAND ===>                                          OUT RECORD:      2
           10        20        30        40        50        60        70    R5
   ----+----|----+----|----+----|----+----|----+----|----+----|----+----|----+
  |  DATE 06/29/98               CAROL'S COOKIES COMPANY           RELEASE 6.2
  |  TIME 13:19:17                      MAIN MENU
  |
 4
  |
  |
  |  ENTER OPTION:
 8
  |
  |          1) PLACE AN ORDER
  |
12          2) CHECK STATUS OF AN ORDER
  |
  |
  |
16
  |
  |
  |
20
```

CA Verify for CICS displays four rows of information above the test stream screen:

- Title
- Command line and record number
- Two-line column number ruler

Also, CA Verify for CICS displays a row number ruler to the left of the screen. The default row and column for Display format are both zero. Use the Up (PF7)**,** Down (PF8)**,** Left (PF10), and Right (PF11) commands to change the starting row and column. You can specify a number or Maximum with these commands; for example, Up 7, Right 20, Down Maximum. If you change the starting row or column, CA Verify for CICS uses the new starting position until you change it or exit from CA Verify for CICS.

Field Format

Field format displays the data stream transmitted to or from the terminal rather than the full screen displayed in Display format. Except on erase/writes, Field format usually does not display all the fields shown in Display format; for example, this format displays only those fields which were modified for read-modified inputs.

The Field format version of the screen is illustrated next.

```
CCC.ORDERAPP.001 ---------  RUN: CURRENT SCREEN  ----------------------13:37:14
ENTER COMMAND ===>                                                        R4

                                                      RECORD NUMBER:        2
TYPE:       OUTPUT             TRAN: XCCC             DATA LENGTH:        266
OPERATION:  ERASE/WRITE        DATE: 06/29/1998       SCREEN SIZE:  24 BY  80
RESPONSE:   00:00:00.002       TIME: 13:18:56.730     TERMINAL:          VV01
WCC: C2: NL/EM/CR PRINT CNTL, KEYBOARD RESTORE
ORDER  ROW  COL  ATTR  LENGTH  ----+----|----+----|----+----|----+----|----+---
 SBA     7   16             0
 IC                         0
 FLD     1    1  PSH        4  DATE
 SF              PS         8  06/29/98
 FLD     1   29  PSH       23  CAROL'S COOKIES COMPANY
 FLD     1   63  PSH        7  RELEASE
 SF              PSM        3  6.2
 FLD     2    1  PSH        4  TIME
 SF              PS         8  13:19:17
 FLD     2   36  PSH        9  MAIN MENU
 FLD     7    1  PSH       13  ENTER OPTION:
 SF              UM         0
 FLD     7   17  PS         1
 FLD    10    9  PSH       17  1) PLACE AN ORDER
 FLD    12    9  PSH       27  2) CHECK STATUS OF AN ORDER
```

Header Information

RUN OPTION

The option on the Run Mismatch Options menu selected for this screen during the last run. This information is omitted if the screen did not have a mismatch or was excluded from the last run, or if the test stream was never run.

TYPE

The type of screen; for example, input or output.

OPERATION

The type of operation; for example, read modified, read buffer, write, erase/write, and so on.

RESPONSE/THINK

For output screens, the response time; for example, the elapsed time in hours, minutes, seconds, and thousandths of seconds since the last terminal input. For input screens, the think time; for example, the elapsed time since the last output.

AID/WCC

For input screens, the attention identifier key pressed to generate input.

For output screens, the write control character:

NL/EM/CR PRINT CNTL

For print operations, specifies a variable line length depending on placement of a new line, end of message, or carriage return orders in the data stream.

80 CHARACTER PRINT

For print operations, specifies an 80-character line.

64 CHARACTER PRINT

For print operations, specifies a 64-character line.

40 CHARACTER PRINT

For print operations, specifies a 40-character line.

KEYBOARD RESTORE

Restores keyboard function.

START PRINT

Initiates a printout operation.

RESETMDT

Resets all modified data tags.

ALARM

Sounds the audible alarm.

TRAN

The transaction associated with the screen.

DATE

The date the test stream was logged. This date may have been updated if the screen was edited or changed by Run Mismatch options 7, 9, 10, or 11, or if the test stream was run with the Comparison Type set to None.

TIME

The time the test stream was logged. This time may have been updated if the screen was edited or changed by Run Mismatch options 7, 9, 10, or 11, or if the test stream was run with the Comparison Type set to None.

RECORD NUMBER

The record number of the screen within the test stream.

DATA LENGTH

The total number of bytes transmitted to or from the terminal. For input screens, this includes three bytes for the AID and cursor position. For output screens, this includes one byte for the WCC.

SCREEN SIZE

The logged screen size.

TERMINAL

The terminal to which the screen was sent, or from which it was received.

CURSOR

For input screens, the cursor position (row and column).

Screen Display

CA Verify for CICS starts a new row on your display for every 3270 order in the screen. If there are no orders, only one row is displayed. Data which follows the order appears under the ruler. Non-3270 data streams are displayed in 40-byte rows.

A common sequence of orders is set buffer address (SBA) followed by start field (SF). To condense the display, CA Verify for CICS merges these orders into an FLD order type.

Each row in the display consists of the following:

ORDER

The 3270 order. This will be blank for the first field if the screen does not begin with an order. The most common orders are summarized next. See the *3270 Component Description* or *Programmer's Guide* for a complete description of valid orders.

SBA

Set buffer address specifies the row and column for the data which follows.

SF

Start field specifies the beginning of a field and its attributes.

IC

Insert cursor specifies cursor positioning in the current row and column.

RA

Repeat to address repeats the first character which follows up to the specified row and column.

PT

Program tab specifies the next unprotected field as the location for the data which follows. If there is any data between the program tab order and the previous order, all fields between the current display position and the destination field are set to nulls.

EUA

Erase unprotected to address sets all unprotected fields from the current display position up to the specified row and column to nulls.

ROW

The row specified by the order.

COL

The column specified by the order.

ATTR

The field attributes specified by a start field order:

U

Unprotected field — can be modified

P

Protected field — cannot be modified

N

Numeric field — only numeric data can be typed

S

Cursor will auto-skip over the field

H

High-intensity field

D

Field can be selected with a selector pen

L

Low-intensity (non-display) field

M

Field with modified data tag which will be transmitted from the terminal even if the operator makes no entry

LENGTH

The length of the data following the order.

DATA

The actual data appears next beneath the ruler.

Scrolling

The default scroll amount is 19 rows for the Up (PF7) and Down (PF8) commands and 40 columns for the Left (PF10) and Right (PF11) commands.

The Up and Down commands affect the display starting with the fifth row; for example, the header row on which Type, Tran, and Data Length information appears. The Left and Right commands affect only the ruler and the data beneath it. You can specify a number or Maximum with these commands; for example, Up 4, Right Maximum.

When data is not displayed beginning with the first character, the number of the first column appears in the ruler.

Hex Format

Hex format segments the data stream as Field format does. A new segment appears for each order for 3270 data streams. Non-3270 data streams are displayed in 40-byte increments.

Three rows are used to display each segment:

- Row 1 contains the character representation
- Rows 2 and 3 contain the hexadecimal representation

The Hex format version of the previous example appears next.

```
CCC.ORDERAPP.001 ---------  RUN: CURRENT SCREEN  ----------------------13:37:20
ENTER COMMAND ===>                                                        R9

OFFSET  LENGTH  ----+----|----+----|----+----|----+----|----+----RECORD:      2
     0       4  B.G?
                C1C6
                217F

     4       1  .
                1
                3

     5       9  . .8DATE
                1441FCCEC
                100D84135

    14      10  .006/29/98
                1FFF6FF6FF
                D006129198

    24      28  . *.8CAROL'S COOKIES COMPANY
                1451FCCDDD7E4CDDDCCE4CDDDCDE
                10CD831963D20366295203647158
F2-ROTATE  F3-END  F5-PRV          F7-UP  F8-DWN  F9-FORMAT  F10-LEFT F11-RIGHT
```

This format provides the offset and length of each segment. Because this example is an output screen, the first byte of the first segment is the WCC. For an input screen, the first three bytes of the first segment would be the AID (one byte) and the cursor position (two byte binary offset relative to zero).

Scrolling

The default scroll amount is 5 segments for the Up (PF7) and Down (PF8) commands and 60 columns for the Left (PF10) and Right (PF11) commands. You can specify a number or Maximum with these commands; for example, Up 3, Right Maximum.

Wait for Missing Output

If the application does not return an output screen when expected, CA Verify for CICS displays the Missing Output menu.

```
TCADS.CCC.ORDERAPP.001-----  RUN: MISSING OUTPUT  ------------EDITED TEST STREAM
 ENTER COMMAND ===>                                                        RN
 TCA0463 - THE SCREEN MAY HAVE BEEN DELETED IN AN EDIT SESSION
 TERMINAL: VV01                                        RECORD NUMBER:    15
 THE RUN FUNCTION IS WAITING FOR AN OUTPUT SCREEN. THE OUTPUT SCREEN MIGHT
 HAVE BEEN ELIMINATED FROM THE APPLICATION. THERE IS NO TRANSACTION ACTIVE
 AT THE TERMINAL.

    1  DISPLAY THE ORIGINAL OUTPUT SCREEN WHICH THE RUN FUNCTION IS WAITING FOR

    2  WAIT ANOTHER 20 SECONDS FOR THE OUTPUT SCREEN TO BE SENT BY THE
       APPLICATION

    3  IGNORE TEMPORARY CHANGE (THE OUTPUT SCREEN IS NOT EXPECTED, BUT RETAIN
       THE ORIGINAL SCREEN IN THE OUTPUT TEST STREAM)

    4  ACCEPT PERMANENT CHANGE (THE OUTPUT SCREEN IS NOT EXPECTED OR WANTED,
       DELETE THE SCREEN FROM THE OUTPUT TEST STREAM)

    SELECT OPTION 2 IF THE OUTPUT MIGHT BE DELAYED DUE TO POOR RESPONSE TIME

    SELECT OPTION 3 OR 4 IF THE APPLICATION HAS CHANGED

 F1-HELP        F3-END
```

The message at the top of the screen always indicates that the Run function is waiting for an output screen that may have been eliminated from the application. The message then indicates one of the following:

- There is no transaction active at the terminal. In this case, you may want to select option 3 or 4.
- Transaction xxxx is active at the terminal. In this case, you probably should select option 2.
- Transaction xxxx is active at the terminal, and has an outstanding terminal read request. In this case, you probably should select option 3 or 4.

Output screens may be missing for many reasons, including:

- Response time is slower than when the test stream was logged
- An input screen was deleted by the Edit function
- The application changed
- The "Welcome to CICS" screen is missing. This message is issued to real VTAM terminals but not to the virtual terminals CA Verify for CICS uses for the Run function.
- Screens from a printer were logged, but the terminal which starts the printer was not logged
- Messages from one terminal to another were logged, but both terminals were not logged

Use the End (PF3) command to end the run or choose one of the following options:

1 DISPLAY THE ORIGINAL OUTPUT SCREEN

Displays the original output screen for which the Run function is waiting. Use the End (PF3) command to return to the Missing Output menu.

2 WAIT ANOTHER nn SECONDS

Directs CA Verify for CICS to wait for nn seconds for the missing screen. Select this option at least once to allow for poor response time unless you are sure the screen is really missing. The number of seconds CA Verify for CICS should wait is determined at installation time.

3 IGNORE TEMPORARY CHANGE

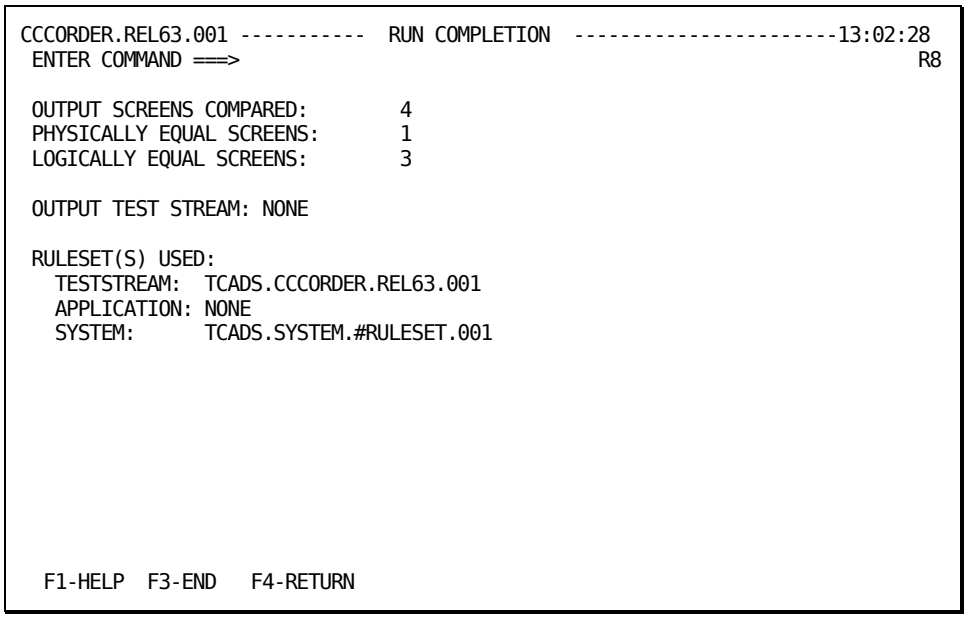
Ignores the missing output screen and continues the run. The original output screen is retained in the output test stream. Select this option if the application will be changed to send the missing screen.

4 ACCEPT PERMANENT CHANGE

Ignores the missing output screen and continues the run. The output screen is deleted from the output test stream. Select this option when the application has changed so that it no longer sends the missing screen.

Complete the Run

When the run completes, CA Verify for CICS displays the Run Completion panel.



Use the End (PF3) command to return to the Run Options menu or the Return (PF4) command to return to the Primary Options menu.

Note: If CA Verify for CICS detects a severe error during the run — for example, an I/O error — it displays an error message instead of the Run Completion menu. In this case, the output test stream, if any, is deleted.

Command List Processor

You can use CA Verify for CICS as a command list processor for any regularly performed series of transactions. This function may be useful for:

- Opening files
- Verifying CICS initialization
- Performing CICS termination tasks

Command list processing uses the Run function to execute an existing test stream. All of the usual Run function menus are suppressed; CA Verify for CICS issues only a completion message.

Create the Test Stream

To use CA Verify for CICS as a command list processor, create a test stream using the following name:

- The default ddname
- An application of CLIST
- A meaningful member name
- The default version

Run the test stream to make sure it executes as expected.

Execute the Command List

To execute the command list, type:

```
XTCA EXEC member
```

where *member* is the member name of the test stream.

CA Verify for CICS runs the test stream and sends one of the following completion messages to your terminal:

- A Run function message, such as, Test Stream Not Found.
- **TCA0603 The Command List Has Been Run, There Were No Unequal Screens** — the normal completion message indicating all of the screens were logically equal.
- **TCA0604 nnnn Unequal Screens, Record Numbers nnnn, nnnn, nnnn, nnnn** — the specified number of screens were not equal. The record numbers of the first four unequal screens are listed.

Execute a Run without Menus

You can execute a run without using any of the Run function menus. Only a completion message is sent to your terminal.

To execute a menuless run, key in the following command. Minimum abbreviations are underlined.

```
XTCA RUN d.a.m.v. THINK(NONE) LIMIT(999999) [SCREEN/LOGICAL/PHYSICAL/NONE] SUPOTS
```

XTCA Run (Required)

Invokes the run.

d.a.m.v. (Required)

Identifies the test stream:

d

DDname

a

Application

m

Member

v

Version

Think(NONE) (Optional)

Specifies no simulated operator think time, the default. You can also specify a number of seconds or a percentage. See the Run Options section for valid parameters.

Limit(nnnnnn) (Optional)

Specifies the maximum number of mismatches CA Verify for CICS should allow. The default is 999999. If this number is exceeded, CA Verify for CICS stops the run. Mismatches generated up to that point are recorded and the output test stream is retained.

Screen/Logical/Physical (Optional)

Specifies the type of comparison. The default is screen. See the Run Options section for an explanation of these options.

Supots (Optional)

The default is for CA Verify for CICS to create an output test stream with the same name as the input test stream and the version incremented to the next available one. Mismatch data is automatically recorded. Specify this option to suppress the output test stream.

Processing during a Menuless Run

During a menuless run, CA Verify for CICS:

- Ignores missing outputs
- Does not perform mismatch confirmation
- Accepts I/O mismatches

When the run completes, CA Verify for CICS returns one of the following messages:

- A Run function message, such as, Test Stream Not Found, or Parameter Is Invalid.
- **TCA0612 nnnn Unequal Screens, Output to output.test.stream** — the specified number of screens was not equal. This message also identifies the name of the output test stream.
- **TCA0613 Mismatch Limit Exceeded, Run Ended Prior to Completion** — the run did not complete because the number of mismatches exceeded the number specified in the Limit option.
- **TCA0617 nnnn Unequal Screens, Output Test Stream Suppressed** — the specified number of screens was not equal. No output test stream was created because you specified the Supots option.

Chapter 6: Browse Function

This section contains the following topics:

Overview

The Browse function displays online the input and output screens captured during the Log and Run functions. You can view:

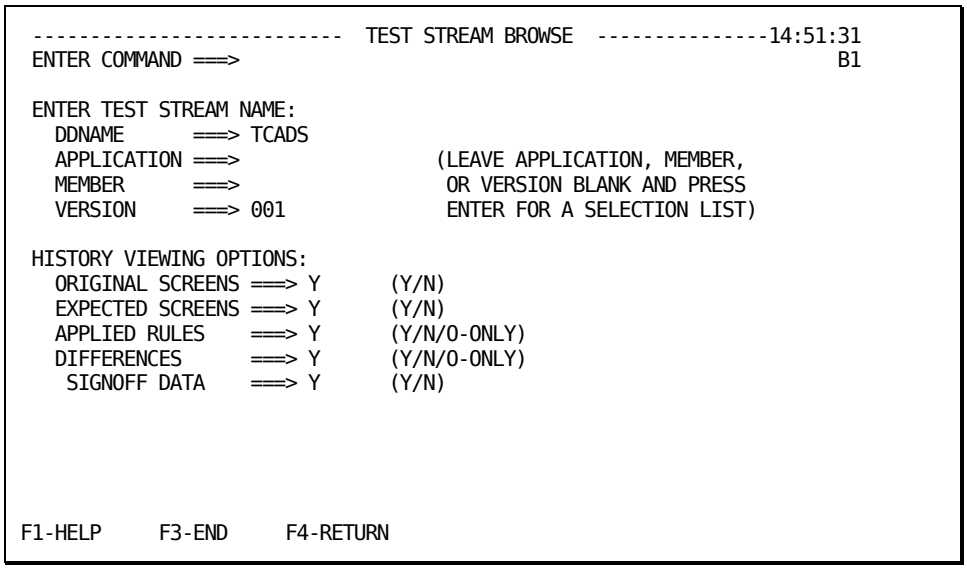
- The original screens captured during logging
- The expected screens created during the Run function
- The current screens captured during the Run function
- The differences CA Verify found between the expected and the current screens
- The rules that were applied to the original screen which created the expected screen
- Operator instructions to CA Verify for CICS during the Run function when CA Verify for CICS detected a mismatch

Use the Browse function to analyze online the results of your testing. Use the Edit function if you intend to modify the test stream; use the batch Print function if you want a hard copy.

Note: How to browse through rulesets and rules (which includes variables) is covered in the chapter "Rules Function".

Invoke the Browse Function

To browse a test stream, type **B** on the Primary Options Menu. CA Verify for CICS displays the Test Stream Browse menu.



The default ddname, Application, and Version values appear; these can be changed.

If you know the test stream you want to browse, follow these steps:

1. Key in the Application name, Member name, and other identifiers, if necessary
2. Optionally modify the History viewing options
3. Press Enter.

   CA Verify for CICS displays the Directory Information menu for that test stream.

If you don't know the test stream you want to browse, follow these steps:

1. Leave the Application and Member field blank.
2. Optionally, blank out other identifiers.
3. Press Enter.

   CA Verify for CICS displays the Test Stream Selection menu.

Select a Test Stream

The Test Stream Selection menu lists all the test streams for the specified ddname.

```
      --------------  BROWSE: TEST STREAM SELECTION  --------------14:00:10
ENTER COMMAND ===>
FILE: TCADS                                                          MULTI-
                          LOG-DATE  BY      RECORDS  RUN-DATE    RESULT   TERM
_ CLIST    OPERATOR 001 01/02/1998 AJC         28    03/21/1998 LGC EQ
                    CEMT INQUIRY OF ALL PROGRAMS AND FILES
s DEMO     DEMOPROG 001 03/22/1998 NGN          8
                    CUSTOMER MAINTENANCE DEMO PROGRAM
_ PAYROLL  DOCUMENT 001 02/01/1998           120
                    ALL PAYROLL SCREENS FOR USER TRAINING
_ PAYROLL  INQUIRY  001 01/16/1998 RSM       154  01/16/1998   NOT EQ
                    TEST OF NEW INQUIRY MENU
_ PAYROLL  UPDATE   002 01/28/1998           112
                    TEST OF NEW UPDATE SCREEN
_ QA       TEST1    003 01/29/1998 NGN      1022  02/29/1998   NOT EQ
                    TEST1 IN CONVERSION FROM DOS  TO MVS
_ QA       TEST2    007 02/21/1998           724
                    TEST2 IN CONVERSION FROM DOS TO MVS
_ SYSTEMS  MIGRATE  008 02/21/1998 SAB      2380  03/14/1998   LGC EQ     M
                    MIGRATION TEST FOR CICS 1.7 TO 2.1
_ SYSTEMS  STRESS1  018 03/22/1998 JSN      1502
                    STRESS TEST FOR CICS TEST REGION
_ SYSTEMS  STRESS2  019 03/22/1998          1208
```

Your position in the list depends on your specifications on the Test Stream Browse menu.

- If both the Application and Member fields are blank, CA Verify for CICS positions you at the first stream for the specified ddname.
- If you fill in a partial name, CA Verify for CICS positions you accordingly. For example, if you specify PAY as the Member name and blank out the Version number, CA Verify for CICS positions you just before the first test stream whose member name begins with 'PAY'.

See the "Run Function" chapter for an explanation of the Test Stream Selection menu.

Type **S** to the left of the test stream you want to browse and press Enter. CA Verify for CICS redisplays the Test Stream Browse menu with the Test Stream identifiers filled in. Specify the History viewing options and press Enter to display the Directory Information menu for the test stream.

Use the Up (PF7) and Down (PF8) commands to scroll through the Test Stream Selection list. To override the default, specify the number of lines or Maximum; for example, Down 25, Down Maximum.

Specify Browse Viewing Options

Complete the Test Stream Browse menu by specifying the Browse viewing options. Type **Y**, **N**, or **O**.

If Record History was set to Y for the run that created this test stream, then original screens, rules (if available), and unequal rows will be included when you Browse.

Note: When you view input only test streams, use either Field or Hex format; do not use Display format.

When you have completed the Test Stream Browse menu, press Enter.

- If you have not specified a complete test stream name, CA Verify for CICS displays the Test Stream Selection list.
- If you have specified a complete test stream name, CA Verify for CICS displays the Directory Information menu for that test stream.

ORIGINAL SCREENS

Y, the default, displays the original logged screens which differ from the current screens, subject to your specifications for the Differences option. **Y** is valid only for test streams which have mismatch information recorded where the Record History option was Y. Type **N** if you do not want to display the original screens.

EXPECTED SCREENS

Y, the default, displays the expected screens. The expected screen is the original screen with rules applied to it. **Y** is only valid if Record History was set to Y during the Run.

APPLIED RULES

Y, the default, lets you view a list of the rules that were applied to each original screen as it was processed. **Y** is only valid for test streams that have Record History set to Y during the Run. Type **N** if you do not want to view the rules; type **O** if you only want to view the screens that have had rules applied to them.

DIFFERENCES

Y, the default, displays the unequal rows associated with the current screens. Type **O** to display only those records for which CA Verify for CICS detected a mismatch. **Y** and **O** are valid only for test streams that have mismatch information recorded where the Record History option was Y. Type **N** if you don't want to display unequal row information.

SIGNOFF DATA

Y, the default, displays the information, if any, the operator supplied during the Run function when CA Verify for CICS detected a mismatch. **Y** is valid only for test streams for which mismatch signoff information was recorded; for example, the Require Signoff Data option was Y. Type **N** if you don't want to access this information.

Browse Directory Information

When you select a test stream, CA Verify for CICS redisplays the Test Stream Browse panel with the test stream name fields completed. Press Enter to display the Browse: Directory Information panel that contains the directory information for the test stream you selected.

```
CCC.ORDERAPP.002 ----  BROWSE: DIRECTORY INFORMATION  -----------------13:40:40
ENTER COMMAND ===>                                                        B3

 DESCRIPTION: CAROL'S COOKIES COMPANY DEMO PROGRAM



            LOG:       RUN:       EDIT:         TEST STREAM:        IN:   OUT:
 INVOKED BY:                                      TOTAL SCREENS:       6      6
 INVOKED ON: 06/02/1998 06/03/1998                AVERAGE BYTES:      46    315
 START TIME: 13:23:04   08:34:50
 DURATION:   00:02:02   00:00:00                LAST RUN:           IN:   OUT:
 SYSTEM:     GRIA5451   GRIA5451                  EQUAL:             6      1
 STATUS:     NORMAL     LOGICALLY EQUAL           EQUIVALENT:                5
 TERMINAL:   A60L2048                             IGNORED:                   0
 VSAM CI'S:  2                                    ACCEPTED:                  0
                                                  INSERTED:          0       0
 AVERAGE THINK TIME:    00:00:20.256              DELETED:           0       0
 AVERAGE RESPONSE TIME: 00:00:00.013              CHANGED:           0
 MAXIMUM SCREEN SIZE:   24 BY  80                 NOT RUN:           0       0
 PROTECTION STATUS:                                          OWNER:
 ORIGINATING TEST STREAM: CCC.ORDERAPP.001      CREATED BY FUNCTION: RUN

 F1-HELP      F3-END       F4-RETURN
```

If you press Enter, CA Verify for CICS displays:

- the Initial Terminal Status menu for a *single* terminal test stream
- the Terminal Selection menu for a *multiple* terminal test stream

If you type a record number in the Command field and press Enter, CA Verify for CICS displays the record you selected.

CA Verify for CICS provides directory information describing Log, Run, and Edit functions involving the test stream. If the Utilities were used to append, insert, or merge the test stream, the Edit heading will be replaced with the appropriate heading.

DESCRIPTION

The description associated with the test stream when it was logged. This description may have been updated by the Edit function or Update utility.

INVOKED BY

The ID of the user who initiated the function.

INVOKED ON

The date the test stream was logged, last run, or edited.

START TIME

The time the function was initiated.

DURATION

The duration of the Log function or the estimated duration of the Run function, based on total response time.

SYSTEM

The CICS job name under which the Log or Run function was initiated.

STATUS

For the Log function:

Normal Incomplete:

CA Verify for CICS could not log some screens because the data capture buffer overflowed or the data set became full

Ext:nnn:

The data stream was extended; nnn indicates the number of records that existed before the extension

For the Run function:

Equal:

The data streams produced by the application were physically equal to those logged

Logically Equal:

The output screens produced by the application were logically equal to those logged, but the data streams were not physically equal

Not Equal:

The output from the application was not the same as the output logged

TERMINAL

The terminal used to log the test stream. For a multiple terminal test stream, the number of terminals appears instead.

VSAM CI'S

The number of control intervals used to store the test stream.

AVERAGE THINK TIME

The average time which elapsed between an output screen and the next input from the terminal.

AVERAGE RESPONSE TIME

The average time which elapsed between terminal input and the next output screen.

MAXIMUM SCREEN SIZE

The largest screen size used in the test stream.

PROTECTION STATUS

The Test Stream Protection option specified when the test stream was logged.

ORIGINATING TEST STREAM

The source of this test stream. In other words, the test stream that was run which created this test stream.

OWNER

The person to whom this test stream belongs.

CREATED BY FUNCTION

The function that was used on the source test stream to create this test stream. Functions can be LOG, RUN, COPY, or RENAME.

Test Stream

The following fields are displayed in the Test Stream area.

TOTAL SCREENS IN: OUT:

The number of input and output screens in the test stream.

AVERAGE BYTES IN: OUT:

The average number of bytes in the input and output data streams.

Last Run

The following fields are displayed if the test stream was run.

EQUAL IN: OUT:

The number of physically or logically equal screens.

EQUIVALENT OUT:

The number of output screens which became logically equal through the definition of variable fields.

IGNORED OUT:

The number of unequal output screens which were ignored (option 6 on the Run Mismatch Options menu).

ACCEPTED OUT:

The number of unequal output screens which were accepted (option 7 on the Run Mismatch Options menu).

INSERTED IN: OUT:

The number of screens which were inserted (options 4 and 5 on the Run Mismatch Out-of-Synch Options menu).

DELETED IN: OUT:

The number of screens which were deleted (options 9 and 10 on the Run Mismatch Out-of-Synch Options menu).

CHANGED IN:

The number of input screens which were changed (option 8 on the Run Mismatch Options menu or a different buffer value when a Read Buffer is issued and the installation option READBUF=CURRENT is in effect).

NOT RUN IN: OUT:

The number of screens which were not run. The terminals associated with these screens were not selected when the test stream was run or you terminated the run before they were processed.

Browse Terminal Information

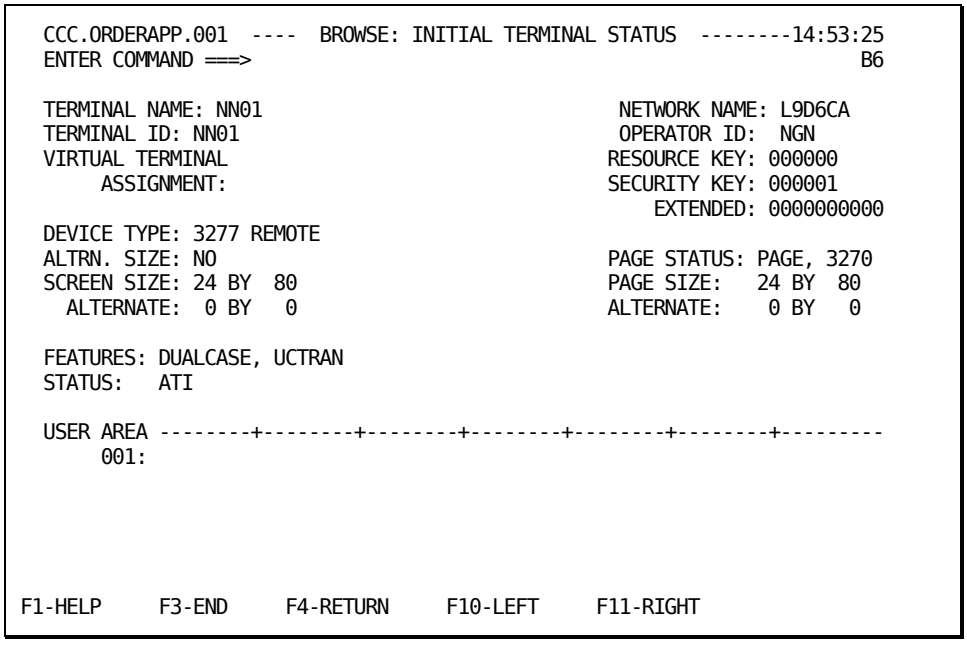
When you press Enter on the Directory Information menu, CA Verify for CICS displays:

- The Initial Terminal Status menu for a single terminal test stream
- The Terminal Selection menu for a multiple terminal test stream

The Initial Terminal Status menu is discussed next. See the Select a Terminal section later in this chapter for a discussion of the Terminal Selection menu.

Display the Initial Terminal Status

The Initial Terminal Status menu displays information for each terminal in the test stream.



This menu displays information about a terminal used to log the test stream. CA Verify for CICS uses this information to synchronize the virtual terminal used in the Run function. Most of the terminal status information is determined by the system programmer when the network is defined.

If the Log function extended a test stream, the screens may be from multiple terminals. In this case, the original terminal status information is used for the entire test stream.

When you have finished browsing the terminal status information for a single terminal test stream, press Enter to display the Record Selection menu.

TERMINAL NAME

Either the terminal network name or the CICS terminal ID, whichever was specified when the terminal was logged.

TERMINAL ID

The CICS terminal ID used to log the test stream.

VIRTUAL TERMINAL ASSIGNMENT

The virtual terminal (if any) assigned to the terminal.

NETWORK NAME

The network name of the terminal used to log the test stream. For non-VTAM terminals, the access method (for example, TCAM, BSAM) is displayed instead.

OPERATOR ID

The ID of the operator who was signed on when the Log function was initiated.

RESOURCE KEY

The resource access key of the operator who was signed on when the Log function was initiated. From right to left, each bit corresponds to a resource level.

SECURITY KEY

The security key of the operator who was signed on when the Log function was initiated. From right to left, each bit corresponds to one of the 24 basic security codes. For example, X'08100F' represents security codes 1, 2, 3, 4, 13, and 20.

EXTENDED SECURITY KEY

The extension of the security key. Additional bits allow security keys 25 - 64.

DEVICE TYPE

The type of terminal used during logging; for example, 3277 local, 3277 remote, and so on.

ALTRN. SIZE

Alternate screen size support:

No

Alternate size is not supported

Yes

Alternate size is supported.

In Use

Alternate size is in use.

Used Last

The last write used the alternate size.

EW/EWA

Needed

Next

An erase/write or an erase/write alternate is needed to change screen sizes.

SCREEN SIZE

The default screen size.

ALTERNATE

The alternate screen size.

PAGE STATUS

Paging status information:

Page

Paging activity is controlled by the terminal user rather than automatically by the paging supervisor.

3270

The paging supervisor will treat the terminal as a 3270.

PAGE SIZE

The default page size for paging operations.

ALTERNATE

The alternate page size.

FEATURES

3270 feature information:

Copy

Copy operations are valid.

Print

The device is a print eligible printer.

Alarm

The terminal has an audible alarm.

Uctran

Input data is translated to upper case

Sel Pen

The terminal has a selector pen.

Dualcase

The terminal has a dual case keyboard.

Local Print

The terminal supports local print.

Mod3A

The terminal has a model 3 printer adapter.

STATUS

Status information:

ATI

The terminal is eligible for automatic transaction initiation.

Dummy

The terminal entry is a dummy for CICS asynchronous transaction processing.

Quiesce

The terminal is quiescing.

Attended

A user is signed on to the terminal.

Rec. Only

The terminal is receive (write) only.

Read Only

The terminal is read only.

Perm Out

The terminal is permanently out of service.

Out of Service

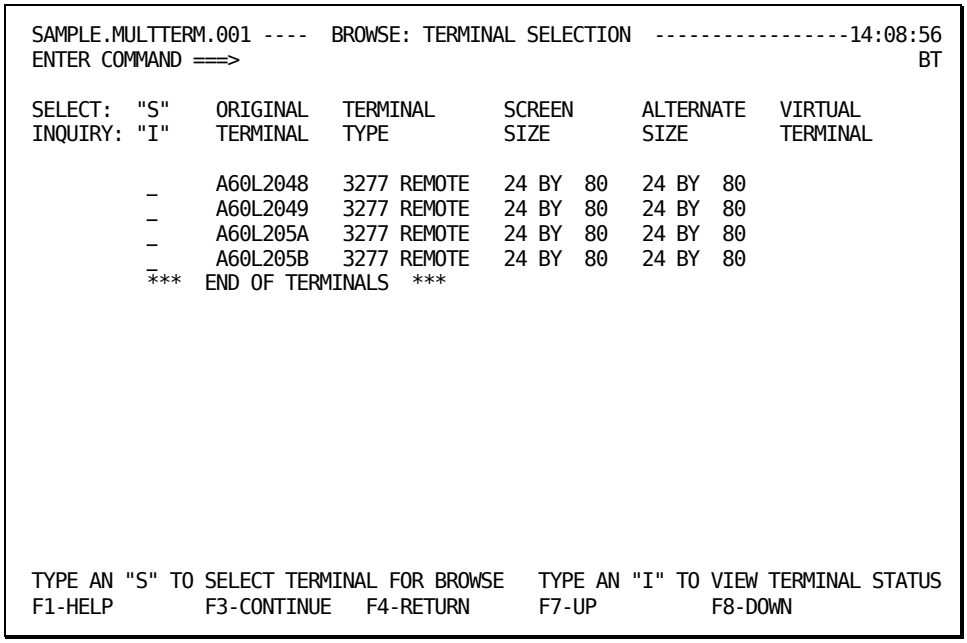
The terminal is out of service.

USER AREA

The terminal user area used by some programs to record current transaction status or to control installation-specific processing options.

Select a Terminal

The Terminal Selection menu is illustrated next. This menu is displayed automatically whenever you run a multi-terminal test stream.



This menu lists all the terminals in the test stream. For each terminal, CA Verify for CICS identifies its name, type, screen size, alternate size, and the virtual terminal (if any) assigned to that terminal.

Key in S to the left of each terminal whose screens you want to browse. Then use the End (PF3) command to display the Record Selection menu.

Key in I to the left of any terminal for which you want terminal status information. Then press Enter to display the Initial Terminal Status menu. When you are finished viewing the terminal status information, use the End (PF3) command to redisplay the Terminal Selection menu.

Use these CA Verify for CICS commands as follows:

Assign

CA Verify for CICS inserts S to the left of each terminal.

Reset

CA Verify for CICS restores the initial screen setting with an underscore to the left of each terminal.

Cancel

CA Verify for CICS cancels the browse and redisplays either the Test Stream Browse menu or the Test Stream Selection menu.

Use the Up (PF7) and Down (PF8) commands to scroll through the terminal list.

Browse Screens

If you select one or more terminals for browsing from the Terminal Selection menu or press Enter from the Initial Terminal Status menu (single terminal test stream only), CA Verify for CICS displays the Record Selection menu. This menu lists the records in the test stream associated with all the terminals you selected.

```
CCC.ORDERAPP.002 --------  BROWSE: RECORD SELECTION  ------------------13:42:05
ENTER COMMAND ===>                                                        B7

   TERMINAL  TRAN  T/R TIME   OP   AID    VIEW  RECORD   ROW: 1 COL: 30
_  A60L2048  XCCC  00:38.057  RM   ENTER            1
_  A60L2048  XCCC  00:00.072  EW            R        2   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:35.706  RM   ENTER            3   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:00.002  EW            R        4   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:12.180  RM   ENTER            5   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:00.002  EW            R        6   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:15.403  RM   PF3              7   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:00.002  EW            R        8   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:01.212  RM   PF3              9   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:00.001  EW            R       10   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:18.979  RM   PF3             11   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:00.002  EW                    12
***  END OF RECORDS  ***




TYPE AN "S" TO SELECT A RECORD
F1-HELP  F3-END  F4-RETURN  F7-UP  F8-DOWN  F9-FORMAT  F10-LEFT F11-RIGHT
```

Key in **S** to the left of the record you want to browse and press Enter. Once you have selected a record, you can use the Prev (PF5) and Next (PF6) commands to display other records. See the "Run Function" chapter for examples of screens in Display, Field, and Hex format.

For each screen, CA Verify for CICS identifies its:

Terminal:

The terminal name

Tran:

The transaction

T/R Time:

The think time (for an input screen) or response time (for an output screen)

OP:

The operation:

W:

Write

EW:

Erase/write

EWA:

Erase/write alternate

RB:

Read buffer

RM:

Read modified

RMA:

Read modified all

EAU:

Erase all unprotected

WSF:

Write structured field

CPY:

Copy

RD:

Read

AID:

Attention ID used to generate input; for example, Enter, PF3

Record:

Record number

View:

Whether there are mismatches (M), rules applied (R), or both (M,R)

ROW COL:

Data from the record which appears in row 1

To position the display beginning with a specific record, type the record number in the Command field and press Enter.

Use the Left (PF10) and Right (PF11) commands to shift the display so you can view additional data from the record.

Use CA Verify for CICS commands as follows:

Profile

Displays the Selection Profile so you can select which information should appear when CA Verify for CICS redisplays the Record Selection menu. This menu is illustrated in the following figure.

Format

Changes the data display beneath the Row/Col to Display or Dump.

See the Browse Commands section later in this chapter for a detailed discussion of the Auto and Format commands.

Change the Fields on the Record Selection Menu

Type the Profile command on the Record Selection menu to display the Selection Profile menu.

```
SAMPLE.MULTTERM.001 -----  BROWSE: SELECTION PROFILE  -----------------14:31:54
 ENTER COMMAND ===>                                                        B8

 S  TERMINAL  (TERMINAL NAME)
 S  TRAN      (TRANSACTION ID)
 S  T/R TIME  (THINK/RESPONSE TIME)
 S  OP        (OPERATION)
 _  LEN       (DATA LENGTH)
 S  AID       (ATTENTION ID)
 S  VIEW      (MISMATCH/RULES INDICATORS)
 _  WCC       (WRITE CONTROL CHARACTER)
 S  RECORD    (RECORD NUMBER)
 _  CURSOR    (CURSOR ROW/COLUMN)
 _  SIZE      (SCREEN SIZE)
 _  DATE      (DATE LOGGED OR LAST CHANGED)
 _  TIME      (TIME LOGGED OR LAST CHANGED)
 S  DATA      (DISPLAY OR DUMP FORMAT DATA)

 TYPE AN "S" TO SELECT A FIELD FOR DISPLAY




 F1-HELP        F3-END        F4-RETURN
```

This menu lists all the fields which can appear on the Record Selection menu. The currently displayed fields have an S next to them. In addition to the fields illustrated in the previous screen, you can select the following fields:

LEN:

Data length

WCC:

Write control character

Cursor:

Cursor's row and column position

Size:

Screen size

Date:

Date the record was logged or last changed by the Edit or Run functions

Time:

Time the record was logged or last changed by the Edit or Run functions

When you have finished selecting fields, use the End (PF3) command to return to the Record Selection menu. The fields you have selected will now appear.

Browse Commands

Use the following commands to browse screens:

Command	Function
Up (PF7)	Scrolls up through a record.
Down (PF8)	Scrolls down through a record.
Left (PF10)	Scrolls left through a record.
Right (PF11)	Scrolls right through a record.
Prev (PF5)	Scrolls backward through a test stream.
Next (PF6)	Scrolls forward through a test stream.
Rotate (PF2)	Rotates between the original and current screens and mismatch information.
Enter	Scrolls forward through a test stream.
Format (PF9)	Shifts between Display, Field, and Hex formats.
Find	Scans a test stream for a character string.
Rfind	Repeats the last Find command.
Reset	Removes the Find or Change parameters specified on the menu.
End (PF3)	Ends the browse of a record and redisplays the Record Selection menu.

Most of the previous commands are discussed in detail in the sections which follow.

Scrolling

UP (PF7), Down (PF8), Left (PF10), Right (PF11)

Scrolls through the record for all three formats. You can type these commands with a number to scroll a specified number of rows, columns, or segments; for example, Up 8, Left 20. You can also specify Maximum to scroll to the top, bottom, extreme left or right of the data; for example, Up Maximum.

Next (PF6), Prev (PF5)

Scrolls forward or backward through the test stream in accordance with the Browse viewing options you selected on the Test Stream Browse menu. The default is one record. You can also specify a number or Maximum with these commands. For example, if record 1 is displayed and you key in Next 4, CA Verify for CICS displays record 5. If you key in Next Maximum, CA Verify for CICS displays the last record.

You can also view a record by typing its record number in the Command field. If the record you specify cannot be viewed, the next or last viewable record will be displayed.

Rotate (PF2)

Scrolls within a record in accordance with the Browse viewing options you selected. For example, you can shift between the original output screen, the current output screen and unequal rows.

Enter

Scrolls within a record, like the Rotate command, and from the last viewable part of one record to the next viewable part of the next record in accordance with the Browse viewing options you selected.

Change Formats

Format (PF9)

Use the Format (PF9) command to shift between Display, Field, and Hex formats. Although parameters aren't required, you can also specify Format xxx, replacing xxx with one of the following:

Display

Immediately shifts to Display format

Field

Immediately shifts to Field format

Hex

Immediately shifts to Hex format

NODisplay

Disables Display format

NOField

Disables Field format

NOHex

Disables Hex format

For example, Format F displays the screen in Field format.

If the format being disabled is the one currently in use, CA Verify for CICS automatically switches to the next available format. If the current format is the only one enabled, CA Verify for CICS displays an error message and does not alter the display.

Note: Do *not* use Display format for input-only test streams.

Locate Data

Find/Rfind

Scans the test stream for a character string. You can specify these commands in all three display formats.

The Find command searches once for the specified value; the Rfind command repeats the previous Find command. Use the Rfind command to locate additional occurrences of the value *without* respecifying it.

When you enter the Find command, CA Verify for CICS inserts the Find parameters on the menu as illustrated next.

```
  CCC.ORDERAPP.001 --------  BROWSE: DISPLAY FORMAT  -------------------14:56:22
  ENTER COMMAND ===>                                                RECORD:
2

  ROW      COLUMN              VALUE
  ===>     ===>       SCREEN ID ===>
  ===>     ===>       SCAN      ===>
             10        20        30        40        50        60        70
B5
     ---------+---------+---------+---------+---------+---------+---------+-----
   | DATE 01/04/98               CAROL'S COOKIES COMPANY               RELEASE
6.2
   | TIME 12:49:58                      MAIN MENU
   |
   4
   |
   |
   |
   8    ENTER OPTIONS:
   |
   |
   |            1) PLACE AN ORDER
  12
   |            2) CHECK STATUS OF AN ORDER
   |
   |
  16
```

The Find command has two parameters: Screen ID and Scan. You can specify values for one or both parameters. If you specify both values, both must be satisfied.

Screen ID

The value specified for this parameter applies to the screen as it appears in Display format. The Row and Column sub-parameters must be specified. CA Verify for CICS will search for the value only in the specified row and column.

Scan

The value specified for this parameter applies to the screen in Field format. The Row and Column sub-parameters are optional. Specify a row without a column to limit the search to a single row; specify a column without a row to limit the search to a single column. Omit both row and column to search the entire screen for the value.

Although the Find and Rfind commands can be entered in Display format, the value specified in the Scan parameter applies to the screen in Field format.

Note: To find orders, specify the Find or Rfind command in Hex format, omitting the Scan parameter row and column.

If CA Verify for CICS finds the string, it displays the record in which it was found; otherwise, the current record is redisplayed along with a message.

Specify the Value

Follow these rules when specifying the Screen ID and Scan values:

- A character value should *not* be specified in single quotes; for example, value.
- A hexadecimal value should be preceded by X and enclosed in single quotes; for example, X'value'.
- Embedded blanks are permitted within a character value; trailing blanks are ignored.

When to Use Screen ID or Scan

A difference exists between the data you view on your terminal and the data transmitted to and from your terminal for the following reasons:

- On *output*, successive transmissions can cumulatively build a terminal display.
- On *input*, only *modified* data is transmitted, not the entire display.

Although CA Verify for CICS captures only the actual transmission, during the Browse, Run, and Edit functions the terminal display is reconstructed.

Use Screen ID to find a screen based on how it appears at a terminal (Display format). Use Scan to find a screen by the actual data transmitted to or from the terminal (Field format).

Remove Parameters

Reset

Removes the Find command parameters from the menu. These parameters are also removed when the Find command is successfully processed or when the parameters are blank.

Chapter 7: Edit Function

This section contains the following topics:

Overview

The Edit function lets you change input and output screens in a test stream. You can reduce your need for the Edit function by using rules the next time you upgrade your test stream or application to a new release.

Use the Edit function to:

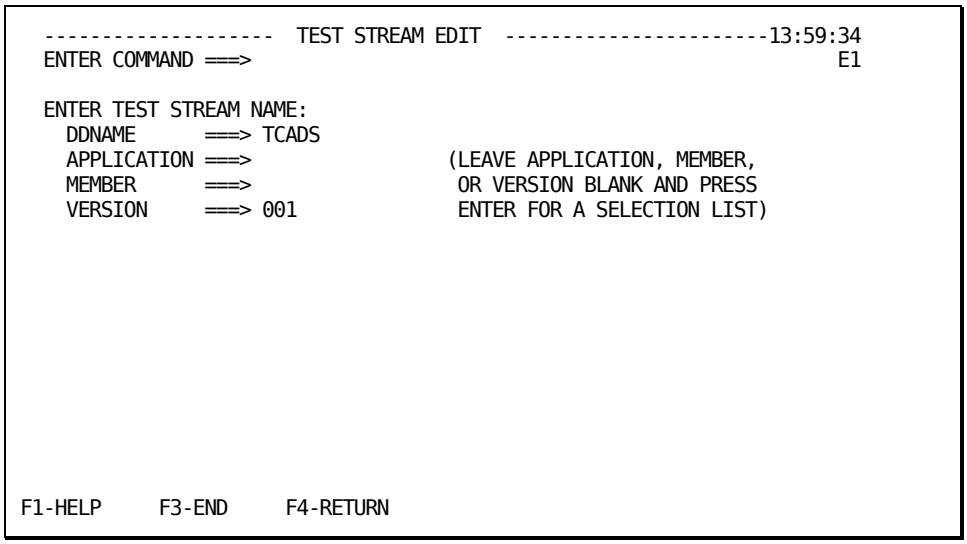
- Change fields, such as dates, before running a test stream
- Create new test data. For example, you can copy a test stream and then revise the input data to create different test cases.
- Eename terminals so they correspond to current terminals. For example, you may want to rename terminals if you are running printers or running with real, rather than virtual, terminals.

If you edit a test stream, CA Verify for CICS also updates the log date and time stored with each record that is changed.

Note: How to edit a ruleset or rule is covered in the chapter "Rules Function".

Invoke the Edit Function

To edit a test stream, type **E** on the Primary Options Menu. CA Verify for CICS displays the Test Stream Edit menu.



The default ddname and Version values appear and can be changed.

If you know the test stream you want to edit, follow these steps:

1. Key in the Application and Member name and other identifiers, if necessary
2. Press Enter.

   CA Verify for CICS displays the Directory Information menu for that test stream.

If you don't know the test stream you want to edit, follow these steps:

1. Leave the Application and Member field blank.
2. Optionally, blank out other identifiers.
3. Press Enter.

   CA Verify for CICS displays the Test Stream Selection menu.

Select a Test Stream from the Test Stream Selection Menu

The Test Stream Selection menu lists all the test streams for the specified ddname.

Your position in the list depends on your specifications on the Test Stream Edit menu.

- If the Application field was blank, CA Verify for CICS positions you at the first test stream for the specified ddname.
- If you left only the Member field blank, CA Verify for CICS positions you just before the first test stream for the specified ddname and Application.
- If you keyed in a partial name, CA Verify for CICS positions you accordingly. For example, if you specified PAY as the Member name without a Version number, CA Verify for CICS positions you just before the first test stream whose member name begins with 'PAY'.

See the "Browse Function" chapter for a description of the Test Stream Selection menu.

Key in **S** to the left of the test stream you want to edit and press Enter. CA Verify for CICS displays the Directory Information menu for that test stream.

Use the Up (PF7) and Down (PF8) commands to scroll through the Test Stream Selection list. The default scroll amount is 17 rows. To override the default, specify the number of lines or Maximum (for example, Down 25, Down Maximum).

Edit Directory Information

The Directory Information menu describes Log, Run, and Edit functions involving the test stream. If the Online Utilities were used to Append, Insert, or Merge the test stream, the Edit heading will be replaced with the appropriate heading.

```
CCC.ORDERAPP.002 -------  EDIT: DIRECTORY INFORMATION  ----------------13:44:46
ENTER COMMAND ===>                                                        E3

 DESCRIPTION ===> CAROL'S COOKIES COMPANY DEMO PROGRAM
             ===>
             ===>

             LOG:       RUN:       EDIT:         TEST STREAM:        IN:   OUT:
 INVOKED BY:                                       TOTAL SCREENS:      6      6
 INVOKED ON: 06/02/1998 06/03/1998                 AVERAGE BYTES:     46    315
 START TIME: 13:23:04   08:34:50
 DURATION:   00:02:02   00:00:00                 LAST RUN:           IN:   OUT:
 SYSTEM:     GRIA5451   GRIA5451                   EQUAL:              6      1
  TATUS:     NORMAL     LOGICALLY EQUAL            EQUIVALENT:                5
 TERMINAL:   A60L2048                              IGNORED:                   0
 VSAM CI'S:  2                                     ACCEPTED:                  0
                                                   INSERTED:           0      0
 AVERAGE THINK TIME:    00:00:20.256               DELETED:            0      0
 AVERAGE RESPONSE TIME: 00:00:00.013               CHANGED:            0
 MAXIMUM SCREEN SIZE:   24 BY  80                  NOT RUN:            0      0
 PROTECTION STATUS:                                              OWNER:
 ORIGINATING TEST STREAM: CCC.ORDERAPP.001      CREATED BY FUNCTION: RUN

 F1-HELP      F3-END       F4-RETURN
```

See the "Browse Function" chapter for a description of the Directory Information fields.

Update Directory Information

You can change the Description field by keying in a new description.

If you edit the test stream — for example, by deleting screens or changing fields, CA Verify for CICS will update the Total Screens and Average Bytes fields when you save the edited test stream. If necessary, CA Verify for CICS will also adjust the Duration, Average Think Time, and Average Response Time fields.

Continue the Edit Function

When you press Enter, CA Verify for CICS displays:

- the Initial Terminal Status menu for a *single* terminal test stream
- the Terminal Selection menu for a *multiple* terminal test stream

If you type a record number in the Command field and press Enter, CA Verify for CICS displays the record you selected—bypassing the Terminal Status and Terminal Selection panels.

Edit Terminal Information

The Initial Terminal Status menu displays information for each terminal in the test stream.

```
 CCC.ORDERAPP.001 ------  EDIT: INITIAL TERMINAL STATUS  ------14:01:14
 ENTER COMMAND ===>                                                 E6
 TERMINAL NAME ===> NN01                          NETWORK NAME: L9D6CA
 TERMINAL ID ===> NN01                             OPERATOR ID: NGN
 VIRTUAL TERMINAL                                 RESOURCE KEY: 000000
 ASSIGNMENT  ===>                                 SECURITY KEY: 000001
                                                      EXTENDED: 000000000
 DEVICE TYPE: 3277 REMOTE
 ALTRN. SIZE: NO                                 PAGE STATUS: PAGE, 3270
 SCREEN SIZE: 24 BY  80                          PAGE SIZE:   24 BY  80
   ALTERNATE:  0 BY   0                          ALTERNATE:    0 BY   0

 FEATURES: DUALCASE, UCTRAN
 STATUS:   ATI

 USER AREA -------+--------+--------+--------+--------+--------+---------
      001:




F1-HELP     F3-END     F4-RETURN     F10-LEFT     F11-RIGHT
```

See the "Browse Function" chapter for a description of the fields.

Use the Enter key to display the Record Selection menu.

Specify two fields on this menu and are explained next.

Terminal Name

Change the terminal by overtyping the terminal name. CA Verify for CICS assumes this is a terminal netname *unless a field mark (X'1E') follows the name.* The field mark identifies the entry as a CICS terminal ID. When you change the terminal name, the appropriate name or terminal ID field reflects the change. The Record Selection menu and Field format displays will reference the old name until you save the edited test stream. For a multiple terminal test stream, the Terminal Selection menu will reference the new name if you redisplay that menu.

Virtual Terminal Assignment

Assign a virtual terminal to be used whenever the test stream is run by specifying a netname in this field. This assignment overrides installation and other specifications, such as running with real terminals or printers. Virtual terminal assignment applies only to VTAM virtual terminals and is ignored if you are using BTAM virtual terminals.

You can specify a generic netname; for example, NET=====. CA Verify for CICS will then use any virtual terminal that fits the specification, such as NET00001, NET00002, and so on.

Select a Terminal

The Terminal Selection menu lists all the terminals in the test stream. See the "Browse Function" chapter for a description of this menu.

Key in **S** to the left of each terminal whose screens you want to edit and use the End (PF3) command. CA Verify for CICS displays the Record Selection menu.

Key in **I** to the left of any terminal for which you want terminal status information and press Enter. CA Verify for CICS displays the Initial Terminal Status menu, illustrated in the previous figure.

Use the following commands:

Assign

CA Verify for CICS inserts **S** to the left of each terminal.

Reset

CA Verify for CICS restores the initial screen setting with an underscore to the left of each terminal.

Cancel

CA Verify for CICS cancels the edit and redisplays either the Test Stream Edit menu or the Test Stream Selection menu.

Up (PF7)/ Down (PF8)

Commands are used to scroll through the terminal list.

Select Records

The Record Selection menu lists the records in the test stream associated with all the terminals you selected and is automatically displayed whenever you select a multi-terminal test stream to browse.

```
CCC.ORDERAPP.002 ----------  EDIT: RECORD SELECTION  ------------------11:21:35
ENTER COMMAND ===>                                                        E7

   TERMINAL  TRAN  T/R TIME   OP   AID    VIEW  RECORD   ROW: 1 COL: 30
_  A60L2048  XCCC  00:38.057  RM   ENTER             1
_  A60L2048  XCCC  00:00.072  EW            R        2   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:35.706  RM   ENTER             3   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:00.002  EW            R        4   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:12.180  RM   ENTER             5   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:00.002  EW            R        6   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:15.403  RM   PF3               7   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:00.002  EW            R        8   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:01.212  RM   PF3               9   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:00.001  EW            R       10   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:18.979  RM   PF3              11   CAROL'S COOKIES COMPANY
_  A60L2048  XCCC  00:00.002  EW                    12
***  END OF RECORDS  ***




TYPE AN "S" TO SELECT A RECORD
F1-HELP  F3-END  F4-RETURN  F7-UP  F8-DOWN  F9-FORMAT  F10-LEFT F11-RIGHT
```

See the "Browse Function" chapter for a description of this menu and a list of the commands which you can enter. See the "Browse Function" chapter for an explanation of the Selection Profile menu which you can use to change the fields displayed on the Record Selection menu.

Key in **S** to the left of the record you want to edit and press Enter.

Screen Formats

The Edit function displays records in Field, Hex, and Display format. Field format is the default. See the discussion of screen formats in the "Run Function" chapter for information on how to use each format.

Note: Do *not* use Display format for input only test streams.

Scroll through the Record

Use the Up (PF7), Down (PF8), Left (PF10), and Right (PF11) commands to display the entire record. The default scroll amounts for each screen format are described in the "Run Function" chapter.

Scroll through the Test Stream

In Display format, you can press Enter to scroll forward from one screen to another in the test stream. For example, if you select the first screen for editing, you can press Enter repeatedly to scroll forward through the entire test stream.

Use the Next (PF6) and Prev (PF5) commands to scroll to the next or previous record. You can specify a number with these commands. For example, if record 1 is displayed and you key in Next 4, CA Verify for CICS displays record 5.

Scroll directly to a record by entering its record number in the Command line. If the specified record cannot be displayed, CA Verify for CICS will display the next or previous record which can be displayed.

Edit Records

You can enter editing commands on any test stream record display. See the Edit Commands section later in this chapter.

When you save a test stream which has been edited, CA Verify for CICS updates the log date and time of each record which has been changed. The date and time are displayed when you view the record in Field format.

Change Screen Data

You can change the data transmitted to or from the terminal in Field or Hex formats. A sample screen in Field format is illustrated next.

Note: Use PF9 to rotate between the formats.

```
CCC.ORDERAPP.002 --------  EDIT: FIELD FORMAT  -----------------------13:48:30
ENTER COMMAND ===>                                                        E4

RUN OPTION: 7                                          RECORD NUMBER:        2
TYPE:       OUTPUT             TRAN: XCCC              DATA LENGTH:        266
OPERATION:  ERASE/WRITE        DATE: 06/29/1998        SCREEN SIZE:  24 BY  80
RESPONSE:   00:00:00.003       TIME: 13:39:44.336      TERMINAL:      A60L2048
WCC: C2: NL/EM/CR PRINT CNTL, KEYBOARD RESTORE
ORDER  ROW  COL  ATTR  LENGTH  ----+----|----+----|----+----|----+----|----+---
 SBA     7   16         00000
 IC                     00000
 FLD     1    1  PSH    00004  DATE
 SF                 PS  00008  06/29/98
 FLD     1   29  PSH    00023  CAROL'S COOKIES COMPANY
 FLD     1   63  PSH    00007  RELEASE
 SF                 PSM 00003  6.2
 FLD     2    1  PSH    00004  TIME
 SF                 PS  00008  13:39:44
 FLD     2   36  PSH    00009  MAIN MENU
 FLD     7    1  PSH    00013  ENTER OPTION:
 SF                 UM  00000
 FLD     7   17  PS     00001
 FLD    10    9  PSH    00017  1) PLACE AN ORDER
 FLD    12    9  PSH    00027  2) CHECK STATUS OF AN ORDER
```

The display is divided into segments determined by each order. For non-3270 terminal test streams, the Field and Hex display is divided into 40-byte segments.

Edit this screen as follows:

- Change the data under the ruler by overtyping it
- Change the length of the data by overtyping the Data Length field. CA Verify for CICS will either truncate the data or pad it with blanks to the specified length.

The following rules apply:

- The row, column, or attribute of a field cannot be changed in Field format; use Hex format to make changes.
- Invalid 3270 characters cannot be changed in Field format; use Hex format to change them.
- Input screens which contain non-display (dark) fields and which were captured when the installation option DRKPROT=YES was in effect can be edited only in Field format.

A sample screen in Hex format appears next.

```
CCC.ORDERAPP.002 --------  EDIT: HEX FORMAT  -------------------------13:49:08
ENTER COMMAND ===>                                                        E9

OFFSET  LENGTH  ----+----|----+----|----+----|----+----|----+----RECORD:      2
     0   00004  B G?
                C1C6
                217F

     4   00001
                1
                3

     5   00009      8DATE
                1441FCCEC
                100D84135

    14   00010   006/29/98
                1FFF6FF6FF
                D006129198

    24   00028    * 8CAROL'S COOKIES COMPANY
                1451FCCDDD7E4CDDDCCE4CDDDCDE
                10CD831963D20366295203647158
F2-ROTATE  F3-END  F5-PRV  F6-NXT  F7-UP  F8-DWN  F9-FORMAT  F10-LEFT F11-RIGHT
```

The display is divided into segments determined by each order. For non-3270 terminal test streams, the display is divided into 40-byte segments.

Edit this screen as follows:

- Change the data, including orders, by overtyping it
- Change the length of the data by overtyping the Length field. CA Verify for CICS will either truncate the data or pad it with blanks to the specified length.
- Change invalid 3270 characters in either of the hex lines, but *not* in the character line

The following rules apply:

- The offset cannot be changed.
- Erased characters in the character line are replaced with blanks; erased characters in hex lines are restored to their original values.
- Changes in either hex line take precedence over changes in the character line, except when a character has been changed in the character line and its corresponding hex representation has been erased in one or both of the hex lines. In this case, the character change takes precedence.

- For a 3270 test stream, if you enter an order or modify existing data so that it becomes an order, the existing field will be split at the new order when the screen is refreshed.
- Input screens which contain non-display (dark) fields and which were captured when the installation option DRKPROT=YES was in effect can only be edited in Field format.

Edit Commands

Use the following commands to edit screens:

Edit Command	Description
Find	Scans a test stream for a character string.
Rfind	Repeats the last Find command.
Change	Scans a test stream for a character string and replaces it with a specified string.
Rchange	Repeats the last Change command.
Reset	Removes the Find or Change command parameters specified on the menu.
Delete	Removes a specified range of screens from a test stream or removes all screens associated with a terminal for a multiple terminal test stream.
Save	Saves the edited test stream. CA Verify for CICS remains in edit mode.
End (PF3)	Saves the edited test stream and redisplays the Test Stream Edit or Test Stream Selection menu.
Return (PF4)	Saves the edited test stream and redisplays the Primary Options Menu.
Cancel	Cancels the edit, discards all changes, and redisplays the Test
Clear	Stream Edit or Test Stream Selection menu.
PA2	Redisplays the current menu. Any data you have keyed in is removed *without* being processed.

Locate Data

Find/Rfind

Scans the test stream for a character string. You can specify these commands in all three formats.

Note: Although the Find and Rfind commands can be entered in Display format, the value specified in the Scan parameter applies to the screen in Field format. Similarly, if the commands are entered in Field or Hex format, the value specified in the Screen ID parameter applies to the screen in Display format.

The Find command searches once for the specified value; the Rfind command repeats the previous Find command. Use the Rfind command to locate additional occurrences of the character string *without* respecifying it.

When you enter the Find command, CA Verify for CICS inserts the Find parameters on the menu as illustrated next.

```
CCC.ORDERAPP.002 --------  EDIT: FIELD FORMAT  ------------------------13:50:02
ENTER COMMAND ===>                                                        E4

ROW      COLUMN              VALUE
===>     ===>      SCREEN ID ===>
===>     ===>      SCAN      ===>
RUN OPTION: 7                                      RECORD NUMBER:        2
TYPE:       OUTPUT             TRAN: XCCC          DATA LENGTH:        266
OPERATION:  ERASE/WRITE        DATE: 06/29/1998    SCREEN SIZE:  24 BY  80
RESPONSE:   00:00:00.003       TIME: 13:39:44.336  TERMINAL:      A55TG001
WCC: C2: NL/EM/CR PRINT CNTL, KEYBOARD RESTORE
ORDER  ROW  COL  ATTR  LENGTH  ----+----|----+----|----+----|----+----|----+---
 SBA     7   16         00000
 IC                     00000
 FLD     1    1  PSH    00004  DATE
 SF                 PS  00008  06/29/98
 FLD     1   29  PSH    00023  CAROL'S COOKIES COMPANY
 FLD     1   63  PSH    00007  RELEASE
 SF                 PSM 00003  6.2
 FLD     2    1  PSH    00004  TIME
 SF                 PS  00008  13:39:44
 FLD     2   36  PSH    00009  MAIN MENU
 FLD     7    1  PSH    00013  ENTER OPTION:
 SF                 UM  00000
```

The Find command has two parameters: Screen ID and Scan. You can specify values for one or both parameters. If you specify both values, both must be satisfied.

Screen ID

The value specified for this parameter applies to the screen as it appears in Display format. The Row and Column sub-parameters must be specified. CA Verify for CICS will search for the value only in that row and column.

Scan

The value specified for this parameter applies to the screen in Field format. The Row and Column sub-parameters are optional. Specify a row without a column to limit the search to a single row; specify a column without a row to limit the search to a single column. Omit both row and column to search the entire screen.

Although the Find and Rfind commands can be entered in Display format, the value specified in the Scan parameter applies to the screen in Field format.

Note: To find orders, specify the Find and Rfind commands in Hex format, omitting the Scan parameter row and column.

If CA Verify for CICS finds the string, it displays the record in which it was found; otherwise, the current record is redisplayed along with a message.

Specify the Value

Follow these rules when specifying the Screen ID and Scan values:

- A character value should not be specified in single quotes; for example, value
- A hexadecimal value should be preceded by X and enclosed in single quotes; for example, X'value'
- Embedded blanks are permitted within a character value; trailing blanks are ignored.

When to Use Screen ID or Scan

A difference exists between the data you view on your terminal and the data transmitted between your terminal and CICS for the following reasons:

- On *output*, successive transmissions can cumulatively build a terminal display.
- On *input*, only *modified* data is transmitted, not the entire display.

Although CA Verify for CICS captures only the actual transmission, during the Browse, Run, and Edit functions the terminal display is reconstructed.

Use Screen ID to find a screen based on how it appears at a terminal (Display format); use Scan to find a screen by the actual data transmitted to or from the terminal (Field format).

Locate and Change Data

Change/Rchange

Finds and changes data in a test stream. These commands can be entered in all three formats. The values specified in the Scan and Replace parameters apply to the screen in Field format. Similarly, if these commands are entered in Field or Hex format, the value specified in the Screen ID parameter applies to the screen in Display format.

The Change command searches for each occurrence of the specified value and replaces it. The Rchange command repeats the previous Change command for the next occurrence. Use the Rchange command to locate and change additional occurrences of the value *without* respecifying it.

You can also specify the Rfind command after the Change command to repeat the search *without* replacing the value.

You can optionally specify the range of records which CA Verify for CICS should search.

If CA Verify for CICS finds the value, it replaces it with the new value and displays the changed record; otherwise, it redisplays the current record along with a message.

If you expand a field, the location of subsequent 3270 fields will not be changed. Avoid destructive field overlap.

When you enter the Change command, CA Verify for CICS inserts the Change parameters on the menu as illustrated next.

```
CCC.ORDERAPP.002 --------  EDIT: FIELD FORMAT  -----------------------08:45:40
ENTER COMMAND ===>                                                      E4

ROW      COLUMN              VALUE
===>     ===>      SCREEN ID ===>
===>     ===>      SCAN      ===>
                   REPLACE   ===>
START WITH RECORD NUMBER ===> 000002  STOP AFTER RECORD NUMBER ===> 000002
RUN OPTION: 1                                          RECORD NUMBER:        2
TYPE:       OUTPUT             TRAN: XCCC              DATA LENGTH:        266
OPERATION:  ERASE/WRITE        DATE: 06/03/1998        SCREEN SIZE:  24 BY  80
RESPONSE:   00:00:00.072       TIME: 08:34:51.345      TERMINAL:      A60L2048
WCC: C2: NL/EM/CR PRINT CNTL, KEYBOARD RESTORE
ORDER  ROW  COL  ATTR  LENGTH  ----+----|----+----|----+----|----+----|----+---
 SBA     7   16         00000
 IC                     00000
 FLD     1    1  PSH    00004  DATE
 SF              PS     00008  06/03/98
 FLD     1   29  PSH    00023  CAROL'S COOKIES COMPANY
 FLD     1   63  PSH    00007  RELEASE
 SF              PSM    00003  6.3
 FLD     2    1  PSH    00004  TIME
 SF              PS     00008  08:34:51
 FLD     2   36  PSH    00009  MAIN MENU
```

The Change command has five parameters: Screen ID, Scan, Replace, Start With Record Number, and Stop After Record Number. The Scan and Replace parameters must be specified; other parameters are optional.

Screen ID

If specified, CA Verify for CICS checks the screen as it appears in Display format before processing the Scan and Replace parameters. The Row and Column sub-parameters *must* be specified. The change will apply only to screens which contain the Screen ID value specified in the row and column.

Usually this parameter is specified with a record range (see the Start With Record Number and Stop After Record Number parameters next).

When processing the Screen ID parameter, CA Verify for CICS treats an attribute or null on the screen as a blank.

Scan

The value specified for this parameter applies to the screen in Field format. You can scan only for data because attributes and nulls will not be changed. (Use Hex format to change attributes and orders.)

The Row and Column sub-parameters are optional. Specify a row without a column to limit the search to a single row; specify a column without a row to limit the search to a single column. Omit both row and column to search the entire screen.

To change orders, the AID, WCC, or cursor position, specify the Change or Rchange command in Hex format, omitting the Scan parameter row and column. Be careful when using these commands in Hex format to avoid changing orders unintentionally.

Replace

Specify the value to replace the Scan value. The number of characters may be the same, greater than, or less than the number of characters in the Scan value. Data in the field following the Scan value will be shifted left or right if the Replace value is shorter or longer than the Scan value.

Start with Record Number/Stop after Record Number

Specify either or both of these parameters to expand the search to a range of records. The default for each parameter is the current record. Use these parameters to make global changes.

See the earlier discussion of the Find/Rfind commands for an explanation of how to specify the value and when to specify Screen ID and Scan.

Remove Parameters

Reset

Removes the Find and Change command parameters from the menu. These parameters are also removed when these commands are successfully processed or when the parameters are blank.

Delete Records and Terminals

Delete

Deletes the displayed record or all the records for a terminal. This command can be entered in all three display formats.

Delete Records

Specify the Delete command to delete the displayed record.

To delete multiple records, key in Delete nn, where nn represents the number of records to be deleted. For example, if you enter Delete 3, CA Verify for CICS will delete the displayed record and the next two records.

For a multiple terminal test stream, the deleted records will be from the same terminal as the displayed record; records from other terminals will be ignored.

Because most applications alternately receive input from the terminal and send output to the terminal, you should usually delete an even number of screens. CA Verify for CICS will display a warning message on the Confirm Delete menu, illustrated in the following figure, if the deletion will result in:

- Two consecutive input screens
- Two consecutive output screens
- An output screen which is the first record for a single terminal test stream or the first record for a terminal in a multiple terminal test stream

Delete Terminals and All Records

To delete *all* the records for a terminal in a multiple terminal test stream, enter the Delete command on the Initial Terminal Status menu. CA Verify for CICS will also delete the initial terminal status information.

You cannot delete all the records for:

- A single terminal test stream
- The only remaining terminal in a multiple terminal test stream. Instead, use the Utilities Delete function to delete the test stream.

Confirm the Deletion

When you specify the Delete command, CA Verify for CICS displays the Confirm Delete menu.

```
-------------------------  EDIT: CONFIRM DELETE  ---------------10:15:30
ENTER COMMAND ===>                                                  ED

 FIRST RECORD TO BE DELETED:          2
 LAST RECORD TO BE DELETED:           2
 NUMBER OF RECORDS TO BE DELETED:     1

NOTE: THE TEST STREAM WILL CONTAIN TWO CONSECUTIVE INPUT SCREENS FROM THE SAME
      TERMINAL. THIS MAY CAUSE SYNCHRONIZATION ERRORS IN THE RUN FUNCTION.
      THE PRIOR  INPUT SCREEN FROM THIS TERMINAL SHOULD ALSO BE DELETED.


F1-HELP     F3-END     F4-RETURN
```

Note: CA Verify for CICS warns you if the deletion might create a synchronization error.

Press Enter to confirm the deletion or use the End (PF3) command to cancel it.

Chapter 8: Rules Function

This chapter explains how to access the Rules function, define rulesets, and specify rules *before* running a test stream. See the "Run Function" chapter for detailed instructions on using rulesets *while* running a test stream.

You can also control what is logged by using logging rules. See the chapter "Log Function" for details on using this type of rule.

Consult the *Rules Primer* for a tutorial that focuses on using the Rules function.

This section contains the following topics:

Overview

CA Verify for CICS is used to identify changes in a test stream of pre-recorded 3270 activities. This test stream can be used over and over again against a new software release for both regression and volume testing. However, instead of using the RUN Mismatch panel to pinpoint these changes during runtime, you can use the Rules function to identify known changes before running a test stream and specify, in advance, how CA Verify for CICS is to handle each change.

To identify changes within a test stream, you must first tell CA Verify for CICS how to recognize the screens that need changes. This is done by creating rules that define what the changes are. This collection of rules is called a *ruleset*.

There are three types of rulesets: test stream, system, and application. How a rule is applied when the test stream is run depends on the type of ruleset that the rule belongs to.

- In a test stream ruleset, rules are applied to all screens in the test stream.
- In a system ruleset, rules are applied to all screens in all test streams in the system.
- In an application ruleset, rules are applied to all test streams in the application.

You can find more detailed information on the different types of rulesets in the section Creating a Ruleset later in this chapter.

All of the rulesets in the CA Verify for CICS data set are listed on the Ruleset Selection Menu. From this menu, you can use the Ruleset Browse feature to examine the summary information about the rules within a ruleset. This information lists all of a ruleset's rules, and summarizes the actions each rule performs.

How is the Rules Function Used?

There are two situations in which rules are used:

- During Logging

  Logging rules are used to control what is logged. Using the Rules function, you can create lists of user IDs, terminals, or transaction IDs that can be included or excluded while logging. For details on this Rules function capability, see the chapter "Log Function".

- During a Run

  Run rules are used to manipulate a test stream for known application changes. You can define these rules during a run or before a run begins.

  A ruleset is run with a particular test stream to dynamically identify specified changes during runtime—allowing your test stream to run more smoothly and efficiently. For example, a common change within an application is to upgrade its release number. Using a ruleset, you can identify this change once, tell CA Verify for CICS how to handle the release number discrepancy whenever it is encountered, and then forget about it. CA Verify for CICS will then interrupt your test stream only for true exceptions.

Procedure for Using the Rules Function

Follow these steps:

1. Review the expected application changes.
2. Determine which of these changes will produce screen changes in your existing test streams (for example, a field was added, a field was moved, a title has been changed, a new screen was added, or an existing screen was removed).
3. Establish a ruleset. This ruleset notifies CA Verify for CICS that you are expecting certain changes and specifies how to handle them.

   To establish a ruleset, you must:

   a. Invoke the Rules function.

   b. Select Add a Ruleset to create a new ruleset.

c. Add rules to the ruleset.

d. Add rule actions to the rules. Recognition criteria and modeling are optional.

While defining rules, calculating the exact coordinates of a field within a panel can be very time consuming. You can allow CA Verify for CICS to automate this calculation for you by using the modeling feature of the Rules function. The modeling feature can be used wherever rule actions are specified. This feature takes the guesswork out of the placement of the fields on a panel by allowing CA Verify for CICS to fill in the placement values for you.

4. Run the test stream. CA Verify for CICS will recognize which screens need to be changed, automatically make the substitutions, and add or delete fields as specified in the rules without interrupting the run.

Create Rules for Existing Test Streams

If you have an existing test stream that was created with Release 4.3 or earlier, you can create a base ruleset for it by either:

- Running the test stream with the current release, or
- Using an online or batch utility copy.

The current release will automatically convert any variables into their corresponding rules and create a base ruleset containing the converted variables, which you can then modify further at your convenience. Complete migration details are in the section Converting Variables to Rules in the chapter "Run Function".

Understand the Terminology

As with most things, the Rules function has its own terminology that you must understand before you can use the function to its full capability. Here is a list of terms, with their explanations, that you will encounter while using the Rules function. Each term is also explained at the point where you would use it.

Expected Screen:

The model screen with the rule actions applied.

Logical Screen:

A collection of test stream records (each reflecting a physical I/O) that together form the screen image seen by the end user.

Model Screen:

A copy of the logical screen that is used to specify field changes.

Recognition Criteria:

A set of identifiers that tells CA Verify for CICS how to recognize a logical screen in order to make changes to it. Recognition criteria can be generic (so it applies to many logical screens), or very specific (so it applies to only one logical screen).

There are two types of recognition criteria:

Field:

The rule looks for a specific field or fields on a screen. Whenever the field is found, the rule is applied.

Screen:

The rule applies only to a specific screen.

Rule:

A rule is a set of recognition criteria and rule actions.

Rule Action:

An instruction for manipulating a field on a screen or which screens to log.

Ruleset:

A set of rules detailing how to handle screens during a run or during a logging session. These rules may contain Recognition Criteria that define, to CA Verify for CICS, which screens these rules should take action on.

Use the Rules Function Panels

There are two types of panels within the Rules function—menus and data entry screens.

- When using a Rules function menu, make your selection as you would when using any other CA Verify for CICS menu. Any rules-specific functions are discussed with the appropriate menu.
- When using Rules function data entry screens, use one of the following methods to complete it:
  - Tab to the input field and type the value directly into the field.
  - Use the modeling feature to let CA Verify for CICS fill in the values for you. How to use the modeling feature is explained in the section Use the Modeling Feature.

When you are finished entering data, you can take one of these actions:

- Press Enter to add the rule action you have just defined.
- Press PF3 (End) to leave the panel without making any changes.

Commands

Use the Caps On/Off command to translate lowercase characters to uppercase characters for screen IDs, value fields, and description fields.

- Caps On, the default, translates characters to uppercase.
- Caps Off ignores translation and uses the capitalization of the characters that appears on the panel.

Common Field Explanations

Fields that appear on many data entry panels are explained next.

RULESET NAME

Identifies the ruleset being defined.

TEST STREAM NAME

Identifies the test stream to which these rules are to be applied. The names of the ruleset and test stream must match.

RULE NAME

The name of this rule.

ROW

Type the number of the row that contains the value to be used. A valid value is any two-digit number beginning with 01 (referring to the first row).

COL

Type the number of the column that contains the value to be used. A valid value is any three-digit number beginning with 001 (referring to the first column).

LEN

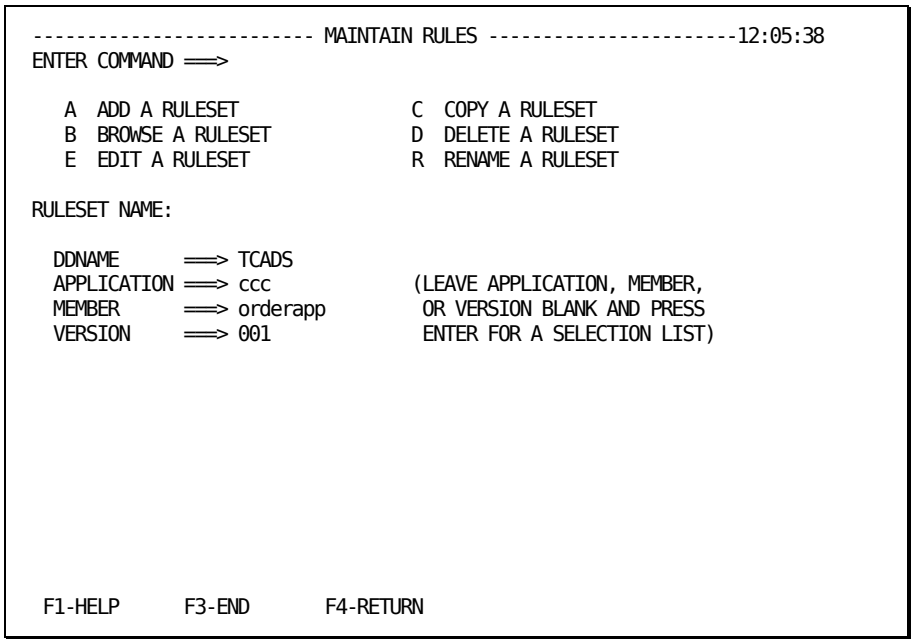
Type the length of the value. Valid values are 1 to 999.

Invoke the Rules Function

Follow these steps:

1. Type M on the Primary Options Menu.

   CA Verify for CICS then displays the Maintain Rules menu. A sample menu is shown next.



2. In the RULESET NAME: area, type the name of the ruleset you want to create or display. Detailed instructions on how to name a ruleset are in the section Name the Ruleset.

3. Indicate which rules activity you want to perform by specifying one of the following letters:

   A

   Add a Ruleset

   B

   Browse a Ruleset

   E

   Edit a Ruleset

   C

   Copy a Ruleset or Rule

   D

   Delete a Ruleset

   R

   Rename a Ruleset

4. Press Enter. CA Verify for CICS displays the menu for the activity you specified.

   Note: If you do not specify a letter, pressing Enter will display a list of existing rulesets.

Browse Through Rulesets and Rules

To display a list of rulesets and rules contained in the data set, type **B** on the Maintain Rules menu. CA Verify for CICS displays the Browse: Ruleset Selection menu shown next.

```
---------------------- BROWSE: RULESET SELECTION  ---------------------13:04:05
ENTER COMMAND ===>
FILE: TCADS
S-SELECT

                      --CREATED-------BY---  --UPDATED-------BY---  RULES  ITEMS
_ A        A        001   09/02/1997                                    1      3
                     A
_ B        B        001   09/02/1997                                    1      2
                         A DIFFERENT APPLICATION
                         A DIFFERENT SET OF RULES TO TRY
_ CCCORDER REL63    001   09/04/1997                                    0      0
                         CAROL'S COOKIES COMPANY
                         CONVERSION FROM REL 6.3 TO REL 7.0
_ DADS34   #RULESET 001   09/02/1997                                    0      0
                         APPLICATION RULESET FOR DADS CONVERSION 3.4 TO 3.5
_ DADS34   SAMP1    001   09/02/1997                                    1      3
                         SAMPLE RECOGNITION/CHANGE TRICK:
                         IF A FIELD HAS A VALUE OF 4, CHANGE IT TO A 5.
_ DADS34   SAMP2    001 09/02/1997                                    1      2
                         ADD A GLOBAL RULE
                         IN A TEST STREAM.
 F1-HELP      F3-END       F4-RETURN    F7-UP        F8-DOWN
```

This menu lists all defined rulesets in alphabetical order by ruleset name. The information shown for each ruleset includes: application, member, version, date the ruleset was created and who it was created by, the date it was updated and who it was updated by, the number of rules the ruleset contains, and the number of rule actions. The ruleset's complete description is displayed on the line(s) beneath the ruleset name.

Invoke the Browse Feature

Follow these steps:

1. Type **S** to the left of the ruleset name on the Browse: Ruleset Selection menu.

   The Rules—Summary (Browse) panel will be displayed. A sample panel is shown next.

```
----------------------BROWSE RULES - SUMMARY ---------------12:53:26
ENTER COMMAND ===>

RULE NAME: RUL00001              RULESET NAME: TCADS.CCC.ORDERAPP.001
LINE   1 TO  14 OF  25        TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

S-BROWSE   P-PREVIEW
  OBJECT      TYPE ROW  COL LEN OP  VALUE FROM MODEL SCREEN/DESCRIPTION
_ RULESET     T/S                   TCADS.CCC.ORDERAPP.001
_  RULE                              RUL00001 - RELEASE 6.2 TO 6.3 CHANGES
_   FLD-RECOG       1   30  23 EQ  CAROL'S COOKIES COMPANY_____________
_    CHANGED  OUT    1   74   1     3_____________________________________
_    VARIABLE OUT    1    7   8     02/11/98______________________________
_    VARIABLE OUT    2    7   8     14:06:31______________________________


 F1-HELP  FF3-END  F4-RETURN  F7-UP  F8-DOWN
```

   For a description of the RULESET NAME, MODEL NAME, and RULE NAME fields, see to the section Use the Rules Function Panels.

   OBJECT

   This column lists the rules within this ruleset and the rule actions within each rule. Values that can appear in the OBJECT column are:

   RULESET

   Ruleset you are viewing

   RULE

   A rule within this ruleset

   FLD-RECOG

   Field recognition criteria

   SCR-RECOG

   Screen recognition criteria

   MOVED

   Moved field

   NEW

   New field

CHANGED

Changed field

VARIABLE

Variable field

DELETED

Removed field

AID KEY

Change in use of a PF key

CURSOR

Change in cursor position

WCC

Change in the Write Control Character

DEL-SCRN

Screen removed

INS-SCRN

Test stream inserted

DATAGEN

Field that includes data manipulation

USERID

User IDs being logged

TRANID

Transaction IDs being logged

TERMID

Terminal IDs being logged

CUT

Data to be copied from one screen and pasted into another

PASTE

Cut data pasted into field

TYPE

Indicates what type of ruleset or rule this is. Valid values are:

T/S

A test stream ruleset

APPL

An application ruleset

SYST

A system ruleset

INCL/EXCL

The rule action contains a list of items to be included or excluded while logging

OUT

The rule action applies to an output screen

IN

The rule action applies to an input screen

The column values for ROW, COL, LEN, OP and VALUE are taken from the panel where that particular ruleset, rule, or rule action was defined.

From this panel you can:

- Select an entry to see the details by typing **S** next to the entry and pressing Enter.
- Preview a rule by typing **P** next to the rule entry and pressing Enter.

Note: Only rules and rule actions can be previewed.

Select an Entry

When you select either a ruleset, rule, or rule action, details about the item you have selected are displayed.

Ruleset

Displays the Browse: Ruleset Directory panel. This panel contains the name of the ruleset, its description, and statistics about it (such as the date it was created and the protection status it has been assigned).

Rule

Displays the Browse Rules—Rule Name panel. This panel contains the name of the ruleset that this rule belongs to, the name of the rule, and its description.

Rule action (like FLD-RECOG, MOVE, or NEW)

Displays the completed data entry screen for the rule action you have selected.

All screens are display-only. To change any of the information, you must return to the Maintain Rules menu (press PF3 until this menu is displayed) and then type E (Edit) on the command line.

Preview a Rule

Use Preview to display what a screen would look like if all of the rule actions in a rule were applied.

Rules summary status messages may appear when you do a preview. See the section Rules Summary Status Messages in the chapter Run Function for a list of messages.

Create a Ruleset

There are three types of rulesets: test stream, application, and system. A test stream can be run with more than one type of ruleset. CA Verify for CICS will try to apply all of the rules from all of the rulesets and will inform you if any conflicts occur.

Note: There is a hierarchy in applying rulesets. During logging, system rulesets are applied first, application next, and test stream rulesets last. During a run, test stream rulesets are applied first, application next, and system rulesets last.

- A *test stream ruleset* is applied only to the test stream whose name matches the ruleset name.

  A test stream ruleset is the most specific. Use it to identify very specific changes for a test stream, such as fields that have moved their position on a screen.

- An *application ruleset* is applied only to those test streams whose application name matches the application name of the ruleset.

  An application ruleset is the next most specific. It is most often used for screens that are common within an application, such as the application main menu. By creating an appplication-level ruleset, changes to the main menu can be identified in one ruleset, but applied to more than one test stream (for example, all test streams whose application node name matches the node name of the application ruleset).

- A *system ruleset* is applied to all test streams in the system. Only one system ruleset can be specified.

  A system ruleset is the least specific. Use it to apply a change to all screens across all applications.

The type of ruleset you create is determined by how it is named; see the section Name the Ruleset for details.

The procedure for creating any type of ruleset is explained in the following table.

Task	Description	Panel Used
1	Name the ruleset	Maintain Rules menu
2	Add a ruleset description	Add Rules—Ruleset Description panel
3	Add rules to the ruleset	Begins with the Add Rules—Rule Actions menu

Name the Ruleset

Follow these steps:

1. Specify a name for the ruleset on the Maintain Rules menu in the RULESET NAME: section.

   The ruleset's name depends on the type of ruleset it is:

   - For a test stream ruleset, the ruleset's name must be the same as the name of the test stream it will be run against.
   - For a system ruleset, the ruleset's name must be:

     ddname.SYSTEM.#RULESET.001

2. Specify the appropriate *ddname*; the rest of the name must be specified as shown.

   - For an application ruleset, the ruleset's name must be:

     ddname.application.#RULESET.001

3. Specify the appropriate *ddname* and *application*; the rest of the name must be specified as shown.

   All ruleset names consist of the following four parts:

   DDNAME

   Must match the ddname in your startup JCL. A DDname can be one to eight characters long, and can be any combination of letters, numbers, or national characters. The first character cannot be a number.

APPLICATION

Must match the application node in the test stream name. An application can be one to eight characters long, and can be any combination of letters, numbers, or national characters. The first character cannot be a number.

MEMBER

Must match the member node in the test stream name. A member can be one to eight characters long, and can be any combination of letters, numbers, or national characters. The first character cannot be a number.

VERSION

Must match the version used in the test stream name. Specify a number from one to three digits long.

4. Press Enter to display the Add Rules—Ruleset Description panel where you can add a description of this ruleset.

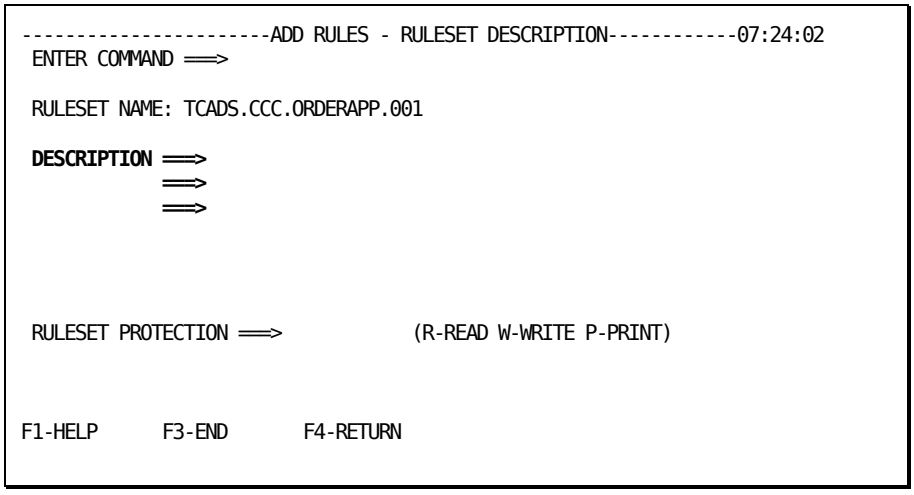
Note: If you do not supply the application, member, or version pieces of the name, the Ruleset Selection menu will be displayed when you press Enter; see the section Browse Through Rulesets and Rules for details.

Add a Ruleset Description

Follow these steps:

1. Type **A** and press Enter on the Maintain Rules menu.

The Ruleset descriptions are added from the Add Rules—Ruleset Description panel.A sample Add Rules—Ruleset Description panel is displayed.



2. Type a description of this ruleset in the DESCRIPTION field.

   Note: The description can be up to three lines long and should be as informative as possible.

   If this ruleset name matches the name of an existing test steam, the test stream description will automatically be copied into the ruleset description. You may leave it or overtype the description with a new one.

Protect a Ruleset

Follow these steps:

1. Specify protection options in the RULESET PROTECTION field on the Add Rules—Ruleset Description panel.

   Valid options are:

   R

   Read. The ruleset cannot be browsed, edited, run, appended to, inserted into, or merged into another ruleset by anyone *except* its owner or a security administrator. If the ruleset is copied or renamed, read protection is extended to the new ruleset.

   W

   Write. The ruleset cannot be extended, edited, renamed, deleted, appended to, inserted or merged into another ruleset, nor can its directory be updated, by anyone *except* its owner or a security administrator.

   P

   Print. The ruleset cannot be printed.

   You can use any combination of options (for example, RW, WP, RWP).

Add Rule to a Ruleset

Rules define what the changes are within an application. As explained previously, a collection of rules is called a ruleset. There are two ways you can invoke a rule:

SCREEN RECOGNITION

Identifies a specific screen that this rule applies to. This type of rule applies changes only to a single specific screen (used when only *one* screen changes). You can have only one Screen Recognition criterion in a rule, but multiple rule actions can be defined for that single screen.

FIELD RECOGNITION

Specifies criteria that a screen must meet before the rule will be applied. This type of rule applies changes only to screens that meet the specified recognition criteria. Field recognition rules are useful when one panel occurs multiple times within a test stream. This type of rule allows you to specify a change once, and then CA Verify for CICS will apply the change to all screens that meet the field recognition criteria.

To add a rule to a ruleset, follow the procedure in the table shown next. All tasks are performed from the Add Rules—Rule Actions panel.

Task	Description
1	Specify a name for the rule
2	Give the rule a description (optional)
3	Identify when this rule should be applied (to a specific screen or to multiple screens based on field recognition criteria). A model screen can be designated at this time.
4	Specify the rule actions

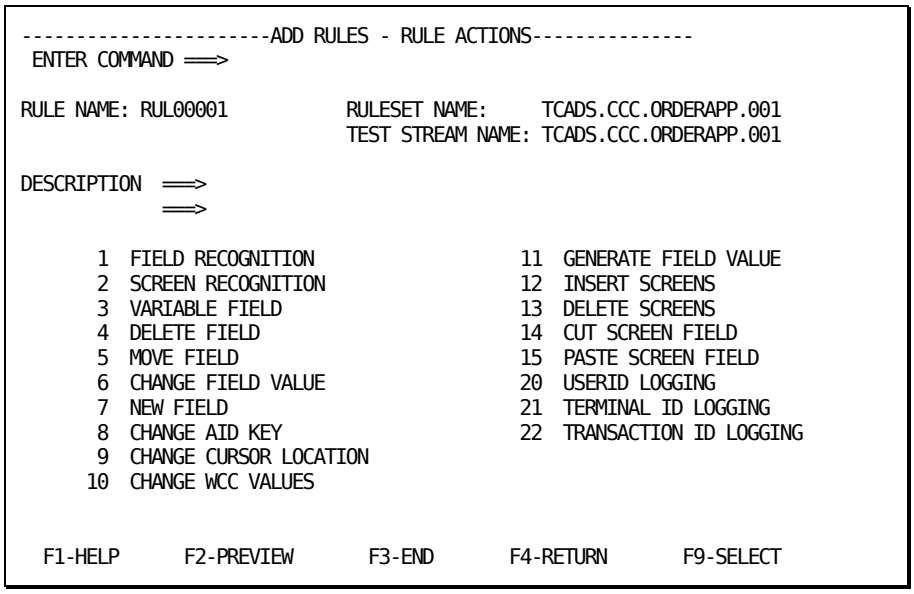
This section explains how to name a rule and specify rule actions. How to locate and select a model screen is covered in the section Select and Use Model Screens.

Name the Rule

Follow these steps:

To specify a rule name, you must complete the RULE NAME and DESCRIPTION fields on the Add Rules—Rule Actions panel.

1. Type A on the Maintain Rules menu, then completing the Add Rules—Ruleset Description panel and pressing Enter.
2. Press Enter and you will be prompted to add a rule action.



RULE NAME

When you first see this screen, a default rule name is given: RUL*nnnnn*, where *nnnnn* is a series of numbers starting with 00001. You can either use the default name, or overtype the name with your own descriptive name. Rule names can be up to eight characters long, and can be any combination of letters and numbers.

DESCRIPTION (optional)

Type a description of the rule, up to two lines in length. The description serves as an explanation of what this rule does, so make it as informative as possible.

Now you are ready to select a model screen and specify the kinds of changes this rule will identify. Selecting a model screen is optional; the next section contains instructions on how to select a model screen.

Select and Use Model Screens

One of the most difficult tasks when specifying changes is figuring out the exact coordinates of the field being changed. The modeling feature lets you easily calculate the exact coordinates of a field within a panel by pointing to the field on a model panel and letting CA Verify for CICS fill in the coordinates on the data entry panel. This feature takes the guesswork out of the placement of the fields on a panel by allowing CA Verify for CICS to fill in the placement values for you.

A *model screen* is a physical representation of the screen being changed. It is used while specifying rule actions to help identify *what* fields are being changed and *how t*hey are being changed. Model screens are selected from the test stream that the ruleset is associated with. The model screen and the test stream automatically become associated by their matching names. Only one model screen can be associated with each rule. However, multiple rules can be created within a ruleset, each with their own model screen.

The collection of test stream records, both input and output, that together form the screen image seen by the end user and which is used to create the model screen, is called a *logical screen*.

Specify a Model Test Stream

The model test stream that you will use depends on the type of ruleset it will be used with.

- For a test stream ruleset, the test stream is used as the model (the default). Here, the model test stream will have the same name as the test stream ruleset.
- For a system ruleset, you must create a model test stream with this name:

  ddname.SYSTEM.#TSTREAM.001

  ddname must match the ddname used in the system ruleset.
- For an application ruleset, you must create a model test stream with this name:

  ddname.*application*.#TSTREAM.001

  ddname and *application* must match the ddname and application node used in the application ruleset.

The use of model test streams and model screens is optional. A model screen can be selected at any time during the rule creation process. If a model has not been selected at the time you want to use it, you will be prompted to select a model screen to use.

The next section explains how to locate and select a model screen. Once you have specified a model test stream and selected a model screen, you are ready to use the modeling feature as explained in the section Use the Modeling Feature.

Locate and Select the Model Screen

The first step in selecting a model screen is locating the logical screen that contains the fields being changed. All of the logical screens within a test stream are listed in the Rules—Record Selection panel shown next. To display this panel, press PF9 (Select) from the Add Rules—Rule Actions panel or from any of the individual rule action panels.

```
CCC.ORDERAPP.001 ---------  RULES: RECORD SELECTION  ------------------11:23:22
ENTER COMMAND ===>                                                        M7

   TERMINAL  TRAN  T/R TIME   OP   AID    VIEW  RECORD   ROW: 1 COL: 30
_  A60L2048  XCCC  00:38.057  RM   ENTER             1
_  A60L2048  XCCC  00:00.013  EW                     2   CAROL'S COOKIES COMPAN
_  A60L2048  XCCC  00:35.706  RM   ENTER             3   CAROL'S COOKIES COMPAN
_  A60L2048  XCCC  00:00.002  EW                     4   CAROL'S COOKIES COMPAN
_  A60L2048  XCCC  00:12.180  RM   ENTER             5   CAROL'S COOKIES COMPAN
_  A60L2048  XCCC  00:00.002  EW                     6   CAROL'S COOKIES COMPAN
_  A60L2048  XCCC  00:15.403  RM   PF3               7   CAROL'S COOKIES COMPAN
_  A60L2048  XCCC  00:00.002  EW                     8   CAROL'S COOKIES COMPAN
_  A60L2048  XCCC  00:01.212  RM   PF3               9   CAROL'S COOKIES COMPAN
_  A60L2048  XCCC  00:00.001  EW                    10   CAROL'S COOKIES COMPAN
_  A60L2048  XCCC  00:18.979  RM   PF3              11   CAROL'S COOKIES COMPAN
_  A60L2048  XCCC  00:00.002  EW                    12
***  END OF RECORDS  ***




TYPE AN "S" TO SELECT A RECORD
F1-HELP  F3-END  F4-RETURN  F7-UP  F8-DOWN  F9-FORMAT  F10-LEFT F11-RIGHT
```

For each screen, CA Verify for CICS identifies its:

Terminal:

The terminal name

Tran:

The transaction

T/R Time:

The think time (for an input screen) or response time (for an output screen)

OP:

The operation:

W:

Write

EW:

Erase/write

EWA:

Erase/write alternate

RB:

Read buffer

RM:

Read modified

RMA:

Read modified all

EAU:

Erase all unprotected

WSF:

Write structured field

CPY:

Copy

RD:

Read

AID:

Attention ID used to generate input; for example, Enter, PF3

VIEW:

Whether there are mismatches (M), rules applied (R), or both (M,R)

Record:

Record number

ROW COL:

Data from the record which appears in row 1.

Note: You can change the columns using the Profile command. See the section Browse Screens in the chapter "Browse Function" for details on Profile.

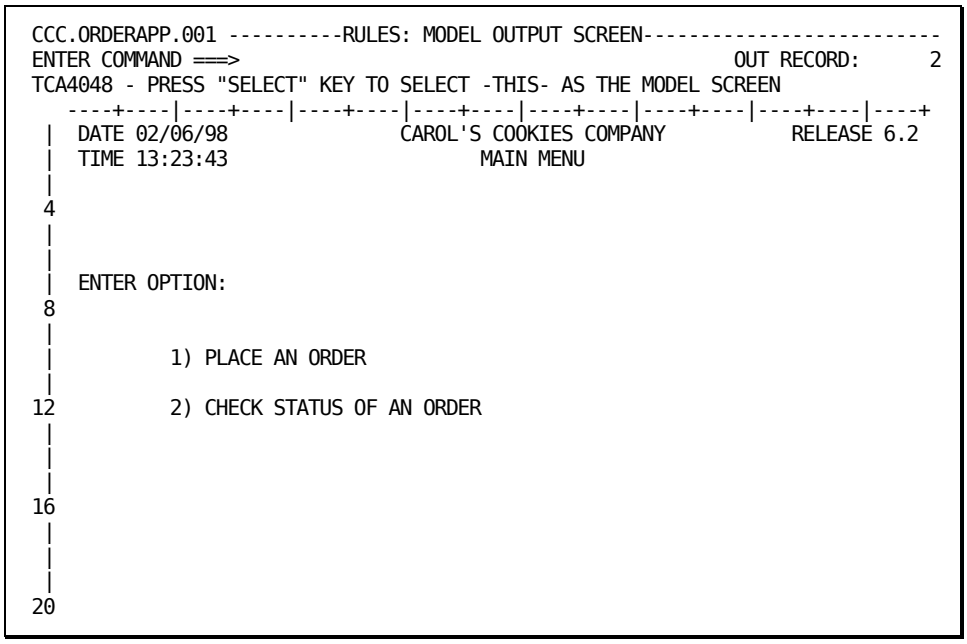
Using the FIND/RFIND Function

Use the Find/RFind function to locate data. First, select a screen from the Rules—Record Selection panel. Then, while viewing the screen, type the Find or RFind command. The Find command searches once for the specified value; the Rfind command repeats the previous Find command. See the section Locate Data in the chapter "Browse Function" for complete details on using Find/RFind.

Note: You can select the first record, so the entire test stream is scanned, or you can position yourself part-way through the list to save scan time.

Confirm Your Choice

The logical screen you have selected or located using the Rules—Record Selection panel is displayed in the Rules—Model Output Screen panel. A sample logical screen is shown next.



To verify that this is the correct screen, view all parts of the screen by scrolling up, down, left, and right.

If the Screen is Correct

If this is the correct screen, press PF9 to select this screen as the model. You will then be prompted to use the model screen to complete your rule actions.

If the Screen is Not Correct

If you had typed search criteria to get to this screen and it is not the correct one, type RFind to display the next occurrence of your search criteria. To enter different search criteria, press PF3 to return to the Rules—Record Selection panel and type new criteria. After you find the correct screen, press PF9 to select it as the model screen.

Use the Modeling Feature

To use the modeling feature, you must have a model screen selected. Since you can select a model screen at any time while defining rules or rule actions by pressing PF9 (Select), it is not necessary to define a model screen before you begin defining rules. How to select a model screen is covered in the section Locate and Select a Model Screen. If you need to *change* a model screen you have already selected, see the section Edit a Rule.

After you have selected a model screen for a rule, it will be automatically displayed whenever you select a rule action to define from the Add Rules—Rule Actions menu for that rule. For example, to define the date as a variable field, select 3 from the Add Rules—Rule Actions menu as shown next.

```
-----------------------ADD RULES - RULE ACTIONS---------------
 ENTER COMMAND ===> 3

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                              TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

DESCRIPTION  ===>
             ===>

       1  FIELD RECOGNITION                  11  GENERATE FIELD VALUE
       2  SCREEN RECOGNITION                 12  INSERT SCREENS
       3  VARIABLE FIELD                     13  DELETE SCREENS
       4  DELETE FIELD                       14  CUT SCREEN FIELD
       5  MOVE FIELD                         15  PASTE SCREEN FIELD
       6  CHANGE FIELD VALUE                 20  USERID LOGGING
       7  NEW FIELD                          21  TERMINAL ID LOGGING
       8  CHANGE AID KEY                     22  TRANSACTION ID LOGGING
       9  CHANGE CURSOR LOCATION
      10  CHANGE WCC VALUES


  F1-HELP      F2-PREVIEW      F3-END      F4-RETURN      F9-SELECT
```

When you press Enter, the model screen you have selected will appear in the Rules: Modified Screen panel, as shown next.

```
CCC.ORDERAPP.001 ------------RULES: EXPECTED SCREEN---------------------------
ENTER COMMAND ===>                                          OUT RECORD:      2
TCA4041 - MOVE CURSOR TO VARIABLE FIELD AND PRESS "SELECT" KEY
   ----+----|----+----|----+----|----+----|----+----|----+----|----+----|----+
  |  DATE 02/06/98              CAROL'S COOKIES COMPANY          RELEASE 6.2
  |  TIME 13:23:43                     MAIN MENU
  |
 4
  |
  |
  |  ENTER OPTION:
 8
  |
  |          1) PLACE AN ORDER
  |
12          2) CHECK STATUS OF AN ORDER
  |
  |
  |
16
  |
  |
  |
20
```

Move the cursor to the first character of the DATE field and press PF9 to select it. The Add Rule Actions—Variable Field panel appears with the field values automatically filled in. A sample is shown next.

```
------------------ ADD RULE ACTIONS - VARIABLE FIELD-----------------08:00:31
 ENTER COMMAND ===>

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                              TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 VARIABLES:
   ROW  COL  LEN   VALUE OR DESCRIPTION
   01   007  008   02/06/98_____________________________________________
                   _____________________________________________________
                   _____________________________________________________


 F1-HELP  F2-PREVIEW  F3-END  F4-RETURN  F9-SELECT
```

If the values are correct, press PF3 (End) to return to the Add Rules—Rule Actions menu where you can continue to define additional rule actions.

Display the Changed Panel

At any time during the rule creation process, you can press PF2 to preview the panel you are working on. All of the changes you have made appear on the panel.

Specify Rule Actions

Rule actions define the changes that are to be made to a panel when this rule is applied. Rule actions include:

- Specifying field or screen recognition criteria
- Identifying a field as a variable (a field that changes from run to run)
- Changes to a field (like changing its length or its place on the panel)
- Adding or removing a field
- Defining a field value as a variable
- Changing the AID or PF keys
- Changing the cursor location or the WCC (Write Control Character) values
- Generating a new value for a date or a field automatically
- Inserting or deleting screens
- Copying data from one screen and pasting it into another screen

Rule action definition begins at the Add Rules—Rule Actions menu; a sample is shown next. To display this menu, press Enter from the Add Rules—Ruleset Description or Add Rules—Confirm Selection panels.

Note: Since rules actions are applied in the order they appear in the ruleset, you should insert the actions in the order you want them applied.

```
----------------------ADD RULES - RULE ACTIONS--------------   08:00:31
 ENTER COMMAND ===>

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                              TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

DESCRIPTION  ===>
             ===>

       1  FIELD RECOGNITION                  11  GENERATE FIELD VALUE
       2  SCREEN RECOGNITION                 12  INSERT SCREENS
       3  VARIABLE FIELD                     13  DELETE SCREENS
       4  DELETE FIELD                       14  CUT SCREEN FIELD
       5  MOVE FIELD                         15  PASTE SCREEN FIELD
       6  CHANGE FIELD VALUE                 20  USERID LOGGING
       7  NEW FIELD                          21  TERMINAL ID LOGGING
       8  CHANGE AID KEY                     22  TRANSACTION ID LOGGING
       9  CHANGE CURSOR LOCATION
      10  CHANGE WCC VALUES


  F1-HELP      F2-PREVIEW      F3-END      F4-RETURN      F9-SELECT
```

For a description of the RULESET NAME, TEST STREAM, and RULE NAME fields, see the section Using the Rules Function Panels.

DESCRIPTION (optional)

Type a description for this rule.

FIELD RECOGNITION

Select option 1, Field Recognition, to identify which fields to find on a screen before this rule can be applied.

SCREEN RECOGNITION

Select option 2, Screen Recognition, to apply this rule only to this screen.

VARIABLE FIELD

Select option 3, Variable Field, to create a new variable field or change an existing one. Variable fields identify fields that CA Verify should ignore in a logical comparison. An example of a field containing a variable value would be date or time.

DELETE FIELD

Select option 4, Delete Field, to remove an existing field from a panel.

MOVE FIELD

Select option 5, Move Field, to change the placement of an existing field on the panel.

CHANGE FIELD VALUE

Select option 6, Change Field Value, to change the value of a field.

NEW FIELD

Select option 7, New Field, to add a new field to the panel.

CHANGE AID KEY

Select option 8, Change AID Key, to change the value assigned to an AID key.

CHANGE CURSOR LOCATION

Select option 9, Change Cursor Location, to change the placement of the cursor on the panel or to make the cursor location a variable.

CHANGE WCC VALUES

Select option 10, Change WCC Values, to change the Write Control Character or to make it a variable.

GENERATE FIELD VALUE

Select option 11, Generate Field Value, if you need to generate or accumulate values for a numerical field or for date aging.

INSERT SCREENS

Select option 12, Insert Screens, if you need to insert logical screens from a test stream.

DELETE SCREENS

Select option 13, Delete Screens, if you need to delete logical screens from a test stream.

CUT SCREEN FIELD

Select option 14, Cut Screen Field, if you need to use some data from this screen on another screen.

PASTE SCREEN FIELD

Select option 15, Paste Screen Field, to place data that was cut from one screen onto this screen.

USERID LOGGING

TERMINAL ID LOGGING

TRANSACTION ID LOGGING

Select one of these options to create a list to include or exclude user IDs (option 20), terminal IDs (option 21), or transaction IDs (option 22) while logging. These selections are discussed in Chapter 4, *Log Function*.

Specify Recognition Criteria

You can specify recognition criteria whenever you define a rule or rule action, but it is not required. There are two types of recognition criteria you can specify:

Field Recognition

Identify which fields must be found on a screen for this rule to be applied.

Screen Recognition

Set this screen as the only screen to which this rule applies.

Each type of recognition criteria is discussed in the following sections.

Note: Field and screen recognition criteria cannot be defined in the same rule. However, each type of criteria can be defined in a separate rule within the same ruleset.

Use Feld Recognition Criteria

To display the Add Rule Actions—Recognition Criteria panel, shown next, type **1** on the Add Rules—Rule Actions menu.

```
--------------------ADD RULE  ACTIONS - RECOGNITION CRITERIA----------07:56:58
 ENTER COMMAND ===>

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                              TEST STREAM NAME: TCADS.CCC.ORDERAPP.001


 ROW COL  LEN  OPER VALUE
 __  ___  ___  __   ____________________________________________________
                    ____________________________________________________
                    ____________________________________________________




  F1-HELP     F2-PREVIEW     F3-END     F4-RETURN     F9-SELECT
```

For instructions on completing a data entry panel and for a description of the RULESET NAME, TEST STREAM, RULE NAME, ROW, COL, and LEN fields, see the section Use the Rules Function Panels.

OPER

Type an operation. Valid values are:

EQ

Equal

NE

Not Equal

LT

Less Than

GT

Greater Than

LE

Less than or Equal to

GE

Greater than or Equal to

VALUE

Type the information you want to match *exactly a*s it appears on the logical screen. A value is required.

Use Screen Recognition Criteria

To display the Add Rules - Screen Recognition panel, type 2 on the Add Rules - Rule Actions menu.

Note: You can have only one Screen Recognition criteria in a rule, but multiple rule actions can be defined for that single screen.

```
---------------------- ADD RULES - SCREEN RECOGNITION -----------08:38:04
 ENTER COMMAND ===>

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                              TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 SCREEN RECOGNITION:
   DESCRIPTION
   ____________________________________________________________________________
   ____________________________________________________________________________
   ____________________________________________________________________________


 THIS RECOGNITION RULE SPECIFIES THAT ALL RULE ACTIONS FOR THIS RULE APPLY
 ONLY TO THE SINGLE SPECIFIC SCREEN SELECTED.




 F1-HELP    F2-PREVIEW    F3-END    F4-RETURN     F9-SELECT
```

For instructions on completing a data entry panel and for a description of the RULESET NAME, TEST STREAM, and RULE NAME fields, see the section Use the Rules Function Panels.

DESCRIPTION

Type a sample of the screen or a short description. Used for documentation purposes only.

Set a Variable Field

Follow these steps:

1. Type **3** on the Add Rules—Rule Actions Menu.

   The Add Rule Actions—Variable Field panel is displayed as shown next.

```
------------------ ADD RULE ACTIONS - VARIABLE FIELD-----------------08:00:31
 ENTER COMMAND ===>

RULE NAME: RUL00001          RULESET NAME:     TCADS.CCC.ORDERAPP.001
                             TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 VARIABLES:
   ROW  COL  LEN   VALUE OR DESCRIPTION
   __   ___  ___   ____________________________________________________
                  ____________________________________________________
                  ____________________________________________________




 F1-HELP  F2-PREVIEW  F3-END  F4-RETURN  F9-SELECT
```

2. Specify the variables.

   For instructions on completing a data entry panel and for a description of the RULESET NAME, TEST STREAM, RULE NAME, ROW, COL, and LEN fields, see the section Use the Rules Function Panels.

3. Type the information about the field to be set as a variable; optional. You can either put a description for why this field is a variable or a sample of the variable data in this field.

A completed Add Rule Actions—Variable Field panel is shown next. The date has been changed to a variable value.

```
------------------ ADD RULE ACTIONS - VARIABLE FIELD------------  ADD SUCCESSFUL
 ENTER COMMAND ===>

RULE NAME: RUL00001          RULESET NAME:     TCADS.CCC.ORDERAPP.001
                             TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 VARIABLES:
   ROW  COL  LEN   VALUE OR DESCRIPTION
   03   068  010   07/10/97_____________________________________________
                    ____________________________________________________
                    ____________________________________________________



 F1-HELP  F2-PREVIEW  F3-END  F4-RETURN  F9-SELECT
```

4. When you are finished specifying variables, press PF3 from the Add Rule Actions—Variable Field panel to return to the Add Rules—Rule Actions Menu.

Delete a Field

Follow these steps:

1. Type **4** on the Add Rules—Rule Actions Menu.

   The Add Rule Actions—Delete Field panel is displayed as shown next.

```
----------------------- ADD RULE ACTIONS - DELETE FIELD -------------08:00:31
 ENTER COMMAND ===>

RULE NAME: RUL00001          RULESET NAME:     TCADS.CCC.ORDERAPP.001
                             TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 VARIABLES:
   ROW  COL  LEN   VALUE OR DESCRIPTION
   __   ___  ___   ____________________________________________________
                    ____________________________________________________
                    ____________________________________________________



 F1-HELP  F2-PREVIEW  F3-END  F4-RETURN  F9-SELECT
```

   For instructions on completing a data entry panel and for a description of the RULESET NAME, TEST STREAM, RULE NAME, ROW, COL, and LEN fields, see the section Use the Rules Function Panels.

   Note: When removing a field that consists of a heading (like birthdate) and an area for data entry, remove both parts of the field by specifying the total length of both parts for LEN.

2. (Optional) Type the name of the field to be deleted in the VALUE OR DESCRIPTION field. You can either explain why this field was deleted or put a sample of the value being deleted in this field.

   A completed Add Rule Actions—Delete Field panel is shown next. The birthdate field is being removed.

```
-------------------- ADD RULE ACTIONS - DELETE FIELD ----------  ADD SUCCESSFUL
 ENTER COMMAND ===>

RULE NAME: RUL00001          RULESET NAME:     TCADS.CCC.ORDERAPP.001
                             TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 VARIABLES:
   ROW  COL  LEN   VALUE OR DESCRIPTION
   15   009  010   BIRTHDATE :_____________________________________________
                   ____________________________________________________
                   ____________________________________________________


 F1-HELP  F2-PREVIEW  F3-END  F4-RETURN  F9-SELECT
```

Move a Field

Follow these steps:

1. Type **5** on the Add Rules—Rule Actions Menu.

   The Add Rule Actions—Move Field panel as shown next.

```
------------------ ADD RULE ACTIONS - MOVE FIELD----------------------07:50:50
 ENTER COMMAND ===>

RULE NAME: RUL00001          RULESET NAME:     TCADS.CCC.ORDERAPP.001
                             TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

   MOVE FROM/TO:
   ROW COL LEN    VALUE OR DESCRIPTION
   __  ___ ___    ____________________________________________________
   __  ___        ____________________________________________________
                  ____________________________________________________


 F1-HELP  F2-PREVIEW  F3-END  F4-RETURN  F9-SELECT
```

2. Type the row, column, and length values of the *current* location of the field being moved on the first line.
3. Type the row and column values of the *new* location for the field being moved on the second line.

4. Type the name of the field to be moved; optional. The field name can be up to three lines long. You can either put an explanation for why this field is being moved or a sample of the data being moved in this field.

   For instructions on completing the data entry panel and description for RULESET NAME, TEST STREAM, RULE NAME, ROW, COL, and LEN fields, see the section Use the Rules Function Panels.

   A completed Add Rule Actions—Move Field panel is shown next. This sample shows a field being moved two spaces to the left.

```
------------------ADD RULE ACTIONS - MOVE FIELD-----------------  ADD SUCCESSFUL
 ENTER COMMAND ===>

RULE NAME: RUL00001          RULESET NAME:     TCADS.CCC.ORDERAPP.001
                             TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

   MOVE FROM/TO:
   ROW COL LEN    VALUE OR DESCRIPTION
   03  065 011   ____________________________________________________
   03  063        ____________________________________________________
                  ____________________________________________________


 F1-HELP  F2-PREVIEW  F3-END  F4-RETURN  F9-SELECT
```

Change a Field Value

Follow these steps:

1. Type **6** on the Add Rules—Rule Actions Menu.

   The Add Rule Actions—Change Field panel is displayed as shown next.

```
--------------------ADD RULE ACTIONS - CHANGE FIELD VALUE-----------08:23:24
 ENTER COMMAND ===>

RULE NAME: RUL00001          RULESET NAME:     TCADS.CCC.ORDERAPP.001
                             TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 CHANGED FIELDS:
 ROW COL  LEN  ATTR FIELD VALUE
 __  ___  ___  ____ ____________________________________________________
                    ____________________________________________________
                    ____________________________________________________

OUTPUT/INPUT FIELD  ===> O      ( I=INPUT, O=OUTPUT)


 F1-HELP  F2-PREVIEW  F3-END  F4-RETURN  F9-SELECT
```

For instructions on completing a data entry panel and for a description of the RULESET NAME, TEST STREAM, RULE NAME, ROW, COL, and LEN fields, see the section Use the Rules Function Panels.

ATTR

Type a field attribute, if the attribute is changing. If the attribute is not changing, leave ATTR blank and CA Verify will default to the original attribute value. Select one of these values for ATTR: U, UM, UH, UHM, UN, UNM, UNH, UNHM, P, PM, PH, PHM, PS, PSM, PSH, PSHM. The meaning of each letter is explained next.

U

Unprotected field; can be modified

P

Protected field; cannot be modified

N

Numeric field; only numeric data can be typed

S

Cursor will auto-skip over the field

H

High-intensity field

M

Field with modified data tag which will be transmitted from the terminal even if the operator makes no entry

FIELD VALUE

Type the new value to be used in the field. A value is required.

INPUT/OUTPUT FIELD

Indicates whether the value being changed is an input or an output field. The default is Output.

A completed Add Rule Actions—Change Field Value panel is shown next. This sample shows the value of a field being changed to 7.0.

```
------------------ADD RULE ACTIONS - CHANGE FIELD VALUE---------  ADD SUCCESSFUL
 ENTER COMMAND ===>

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                              TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 CHANGED FIELDS:
 ROW COL  LEN  ATTR FIELD VALUE
 02  075  003  ____ 7.0_____________________________________________________
                   ____________________________________________________________
                   ____________________________________________________________

OUTPUT/INPUT FIELD  ===> O      ( I=INPUT, O=OUTPUT)


 F1-HELP  F2-PREVIEW  F3-END  F4-RETURN  F9-SELECT
```

Add a New Field

Follow these steps:

1. Type **7** on the Add Rules—Rule Actions Menu to display the Add Rule Actions—New Field panel. A sample panel is shown next.

```
-----------------------ADD RULE ACTIONS - NEW FIELD----------------08:28:40
 ENTER COMMAND ===>

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                              TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 NEW FIELDS:
 ROW COL  LEN  ATTR FIELD VALUE
 __  ___  ___  ____ ____________________________________________________
                   ____________________________________________________________
                   ____________________________________________________________


 F1-HELP  F2-PREVIEW  F3-END  F4-RETURN  F9-SELECT
```

2. For instructions on completing a data entry panel and for a description of the RULESET NAME, TEST STREAM, RULE NAME, ROW, COL, and LEN fields, see the section Use the Rules Function Panels.

ATTR

Type a field attribute, if the field is new. However, leave ATTR blank if the new field is being inserted into the middle of an existing field.

If the field is not new, leave ATTR blank and CA Verify will use the original attribute value for the whole field.

Select one of these values for ATTR: U, UM, UH, UHM, UN, UNM, UNH, UNHM, P, PM, PH, PHM, PS, PSM, PSH, PSHM. The meaning of each letter is explained next.

U

Unprotected field; can be modified

P

Protected field; cannot be modified

N

Numeric field; only numeric data can be typed

S

Cursor will auto-skip over the field

H

High-intensity field

M

Field with modified data tag which will be transmitted from the terminal even if the operator makes no entry

FIELD VALUE

Type the new value to be used in the field. A value is required.

A completed Add Rule Actions—New Field panel is shown next.

```
--------------------ADD RULE ACTIONS - NEW FIELD--------------  ADD SUCCESSFUL
 ENTER COMMAND ===>

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                              TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 NEW FIELDS:
 ROW COL  LEN  ATTR FIELD VALUE
 03  074  002  ____ 19_____________________________________________________
                   _____________________________________________________
                   _____________________________________________________


 F1-HELP  F2-PREVIEW  F3-END  F4-RETURN  F9-SELECT
```

Change the AID Key

Follow these steps:

1. Type **8** on the Add Rules—Rule Actions Menu to display the Add Rule Actions—Change AID Key panel. A sample panel is shown next.

   Note: A rule can contain only one rule action of this type.

```
 -------------------ADD RULE ACTIONS - CHANGE AID KEY--------------13:17:43
 ENTER COMMAND ===>

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                              TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

  ORIGINAL   CHANGE  |  ORIGINAL   CHANGE |  ORIGINAL   CHANGE  |
   AID:      TO      |  AID        TO:    |   AID       TO:     |
                     |                    |                     |
   PF1  ==>  _____   |  PF11 ==>  _____   |   PF21 ==>  _____   |
   PF2  ==>  _____   |  PF12 ==>  _____   |   PF22 ==>  _____   |
   PF3  ==>  _____   |  PF13 ==>  _____   |   PF23 ==>  _____   |
   PF4  ==>  _____   |  PF14 ==>  _____   |   PF24 ==>  _____   |
   PF5  ==>  _____   |  PF15 ==>  _____   |   ENTER==>  _____   |
   PF6  ==>  _____   |  PF16 ==>  _____   |   PA1  ==>  _____   |
   PF7  ==>  _____   |  PF17 ==>  _____   |   PA2  ==>  _____   |
   PF8  ==>  _____   |  PF18 ==>  _____   |   PA3  ==>  _____   |
   PF9  ==>  _____   |  PF19 ==>  _____   |   CLEAR==>  _____   |
   PF10 ==>  _____   |  PF20 ==>  _____   |


  F1-HELP      F2-PREVIEW     F3-END       F4-RETURN     F9-SELECT
```

2. Type the ruleset name, test stream, and rule name in the RULESET NAME, TEST STREAM, and RULE NAME fields.

   ORIGINAL AID

   Contains the original AID key name.

   CHANGE TO

   Type the new value next to the AID keys you wish to change.

   Note: The Add Rule Actions—Change AID Key panel only changes the AID key keystroke. To change the words at the bottom of a panel that indicate PF key use, use the "change field value" option on the Add Rule Actions menu.

Change the Cursor Location

Follow these steps:

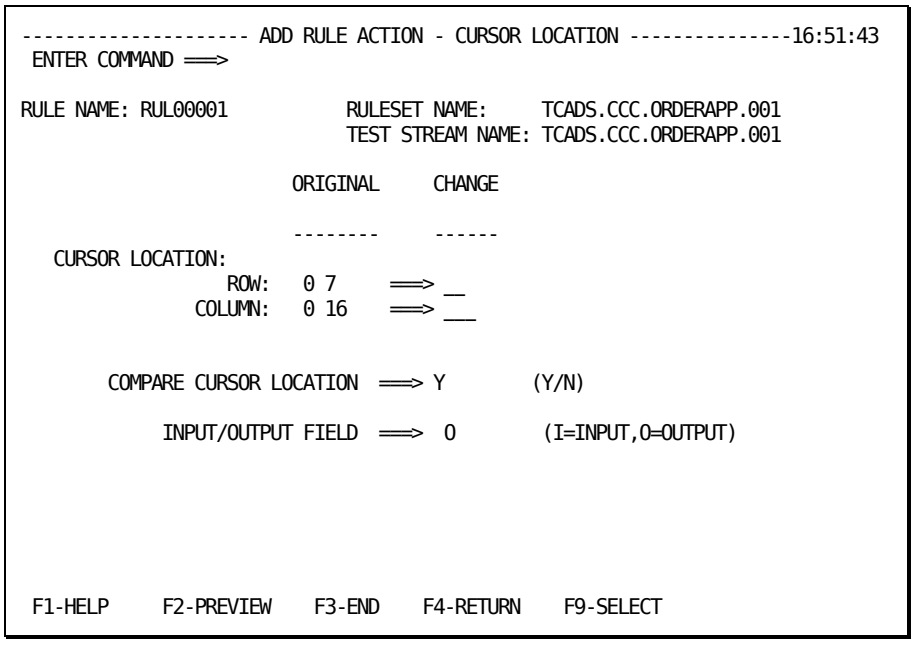
1. Type **9** on the Add Rules—Rule Actions Menu to display the Add Rule Action—Cursor Location panel.

   This panel has two purposes:

   - To identify a new location for the cursor and then compare the cursor's location to make sure it is the correct one, or
   - To set the cursor as a variable so that no comparison is done on the cursor location.

   Note: A rule can contain a maximum of one input cursor change and one output cursor change.

A sample panel is shown next.



2. For a description of the RULE NAME, TEST STREAM NAME, and RULESET NAME fields, see the section Use the Rules Function Panels.

CURSOR LOCATION

The current value for ROW and COLUMN are listed in the ORIGINAL VALUE column. Type the new values for ROW and COLUMN in the CHANGE TO column.

Note: A change in cursor location is only valid if you specify Y for the COMPARE CURSOR LOCATION field.

COMPARE CURSOR LOCATION

Indicates whether you want to compare the location of the cursor. Valid values are Y (yes) or N (no). Specifying N means that no comparison will be done, making the cursor location a variable.

OUTPUT/INPUT FIELD

Indicates whether the criteria specified on this panel applies to the output or input version of the screen. Valid values are I (input) or O (output).

Note: If the criteria apply to both output and input versions of the screen, you must create two rule actions—one for output and one for input.

Change the WCC Value

Follow these steps:

1. Type **10** on the Add Rules—Rule Actions Menu to display the Add Rule Actions—WCC Values panel.

   This panel has two purposes:

   - To identify changes to the setting of the WCC and then compare the WCCs during the Run function, or
   - To set the WCC as a variable so no comparison will be done during the Run.

   Note: A rule can contain only one rule action of this type.

   A sample panel is shown next.

```
 -------------ADD RULE ACTIONS - WCC VALUES---------                13:17:50
 ENTER COMMAND ===>

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                              TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

                         ORIGINAL     CHANGE
                         VALUE        TO
                         --------     ------
            START PRINT:   N      ===> N      (Y/N)
            SOUND ALARM:   N      ===> N      (Y/N)
          RESTORE KYBD:    Y      ===> Y      (Y/N)
              RESET MDT:   N      ===> N      (Y/N)

              COMPARE WCC VALUES  ===> Y      (Y/N)


  F1-HELP   F2-PREVIEW  F3-END   F4-RETURN   F9-SEELCT
```

2. For a description of the RULESET NAME, TEST STREAM, and RULE NAME fields, see the section Use the Rules Function Panels.

   WCC

   The current value for START PRINT, SOUND ALARM, RESTORE KYBD, and RESET MDT are listed in the ORIGINAL VALUE column. Type the new values in the CHANGE TO column. Valid values are Y or N.

   Note: A change in WCC is only valid if you specify Y for the COMPARE WCC VALUES field.

   COMPARE WCC VALUE

   Indicates whether you want to compare the value of the WCC. Valid values are Y (yes) or N (no). Specifying N means that no comparison will be done, making the WCC a variable.

Generate a Screen Value

Follow these steps:

1. Type **11** on the Add Rules—Rule Actions Menu to display the Add Rule Actions—Data Generation panel.

   Note: Use this panel to generate a value for an input or an output field. The target field must be a number, numeric currency, or a numeric date.

   A sample panel is shown next.

```
 ------------------   ADD RULE ACTIONS - DATA GENERATION--------------16:51:53
 ENTER COMMAND ===>

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                              TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 SCREEN LOCATION:
   ROW  COL  LEN  VALUE
     __  ___  ___  ____________________________________________________________
                   ____________________________________________________________
                   ____________________________________________________________

 INITIAL VALUE        ===> N             (Y/N)
 OPERATOR             ===> _             (ADD=+,SUBTRACT=-,MULTIPLY=*,DIVIDE=/)
 OPERATOR VALUE       ===> _______       (1 - 9999999)
 OVERFLOW             ===> N             (Y/N)
 ACCUMULATE           ===> N             (Y/N)
   SCOPE              ===> _             (S=SYSTEM,T=TERMINAL,U=USER)
 DATE AGING FORMAT    ===> _____________
   AGE DATE BY D|Y    ===> _             (D=DAYS,Y=YEARS)

 INPUT/OUTPUT FIELD   ===> I             (I=INPUT,O=OUTPUT)

 F1-HELP     F2-PREVIEW    F3-END    F4-RETURN    F9-SELECT
```

2. For a description of the RULE NAME, TEST STREAM NAME, RULESET NAME, ROW, COL, and LEN fields, see the section Use the Rules Function Panels.

   SCREEN LOCATION

   Indicates the location of the field that contains the value being accumulated or the date being aged. You can fill in the location values by either using a model screen, or by typing the values directly into the appropriate columns.

   Note: The field containing the value being accumulated or the date being aged can be either an input field or an output field.

   INITIAL VALUE

   Indicates whether or not you will supply a starting value for the generated field. Specify N (no) to use the value in the specified field as the initial value. Specify Y (yes) to supply a different starting value; type the new value to be used in the VALUE field.

OPERATOR

Specifies the arithmetic operation to be performed on the initial value; a value is required. Valid values are: + (add the operator value to the initial value), - (subtract the operator value from the initial value), * (multiply the initial value by the operator value), or / (divide the initial value by the operator value).

OPERATOR VALUE

Type a whole number that is to be added to, subtracted from, multiplied by, or used as the divisor of the initial value; a value is required. Valid values are 1 to 9,999,999.

OVERFLOW

Indicates whether processing is to continue if a value overflows the target field length. Specify Y (yes) to ignore any overflows and continue processing. Specify N (no) if overflowing the field boundaries is not allowed; processing will stop and the Mismatch panel will be displayed.

ACCUMULATE

Indicates whether or not you want to use the last generated value for this field as the basis for the next generated value for this field. Type N (no) if you do not, Y (yes) if you do.

If you specify Y, then a value for SCOPE is required. SCOPE controls how the data is accumulated: by user, by terminal, or by system.

Note: If a data generation rule action is changed during a run and the accumulator is set to Y, then the accumulator will not be adjusted for the remainder of the run. The only exception is if the length of the value has changed.

SCOPE

When Y is specified for ACCUMULATE, you must indicate how the data is to be accumulated. Specify S (system) to accumulate the data by system; specify T (terminal) to accumulate the data by terminal; specify U (user) to accumulate the data by user.

DATE AGING FORMAT

Specify the date format you are using in this application. This value is required if the initial value or the value found at the screen location is a date. Valid date formats are:

```
MMDDYY   MM/DD/YYMM / DD / YY     YYDDD
YYMMDD   YY/MM/DDYY / MM/ /DD
YYDDMM   YY/DD/MMYY / DD / MM
```

In any of the previous formats, the year can be specified as two characters (YY) or four characters (YYYY). You can also replace the / with any national character (such as a -). However, the format, the format's length, and the separator characters specified in the date aging format must match what is found in the screen location during the run. If they do not match, a conflict will occur and the Mismatch panel will appear.

If you specify a value for DATE AGING FORMAT, then you must also specify a value for AGE DATE BY.

AGE DATE BY D|Y

Specify how the date is to be aged: by day (D) or by year (Y).

INPUT/OUTPUT FIELD

Indicates whether this rule applies to an input or an output screen. Specify I for input; O for output.

Insert Screens

Follow these steps:

1. Type **12** on the Add Rules—Rule Actions Menu to display the Add Rule Actions—Insert Screens panel.

   A sample panel is shown next.

   Note: A rule can contain only one rule action of this type.

```
 ------------------ADD RULE  ACTIONS - INSERT SCREENS--------------13:18:03
 ENTER COMMAND ===>

RULE NAME: RUL00001          RULESET NAME:     TCADS.CCC.ORDERAPP.001
                             TEST STREAM NAME: TCADS.CCC.ORDERAPP.001


 INSERT SCREENS FROM TEST STREAM:

   DDNAME      ===> TCADS            (LEAVE APPLICATION, MEMBER,
   APPLICATION ===>                  OR VERSION BLANK AND PRESS
   MEMBER      ===>                  ENTER FOR A SELECTION LIST.)
   VERSION     ===>









  F1-HELP      F2-PREVIEW     F3-END       F4-RETURN     F9-SELECT
```

   For a description of the RULESET NAME, TEST STREAM, and RULE NAME fields, see the section Use the Rules Function Panels.

Insert a New Screen

Before you can insert a new screen, you must build a test stream containing the screens you want to insert.

Follow these steps:

1. Give the test stream a name.
2. Record the new screens using the Log function of CA Verify for CICS.
3. Edit the test stream so that only the new records are in it.

Once you have performed these steps, you are ready to complete the field under Insert Screens From Test Stream on the Add Rule Actions—Insert Screens panel.

Type the application, member, and version of the test stream containing the new screens in the appropriate fields. The inserted test stream must be from the same TCADS file as the original test stream. The new screens will be inserted in the existing test stream immediately *after* the logical screen you are currently working on.

Delete Screens

Follow these steps:

1. Type **13** on the Add Rules—Rule Actions Menu to display the Add Rule Actions—Delete Screens panel. A sample panel is shown next.

   Note: A rule can contain only one rule action of this type.

```
 ------------------ADD RULE  ACTIONS - DELETE SCREENS-------------13:18:03
 ENTER COMMAND ===>

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                              TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 DELETE __ SCREENS STARTING WITH THE CURRENT OUTPUT









  F1-HELP      F2-PREVIEW     F3-END       F4-RETURN       F9-SELECT
```

For a description of the RULESET NAME, TEST STREAM, and RULE NAME fields, see the section Use the Rules Function Panels.

Delete Screens when Comparison Type = Screen on Run Option Menu

Type the number of screens to be deleted in the DELETE field. The removal takes place beginning with the current output record. When a screen is deleted, all of the output records that make up the output screen, as well as all of the subsequent input records until the next output screen is encountered, are removed.

Delete Records when Comparison Type = Logical on Run Option Menu

Type the number of records to be deleted in the DELETE field. The removal takes place beginning with the current output record

Copy Data From a Screen

Follow these steps:

1. Type **14** on the Add Rules—Rule Actions Menu to display the Add Rule Actions—Cut Fields panel.

   A sample panel is shown next.

   Note: You can use the data you select any number of times on any number of panels.

```
 ------------------------ADD RULE ACTIONS - CUT  FIELDS----------------16:52:11
 ENTER COMMAND ===>

RULE NAME: RUL00001          RULESET NAME:     TCADS.CCC.ORDERAPP.001
                             TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 CUT FROM:
   ROW  COL  LEN   VALUE OR DESCRIPTION
   __   ___  ___   ____________________________________________________________
                  ____________________________________________________________
                  ____________________________________________________________

 OUTPUT/INPUT FIELD  ===> O             (I=INPUT,O=OUTPUT)


 CUT ID = C00001




F1-HELP    F2-PREVIEW    F3-END    F4-RETURN     F9-SELECT
```

2. For a description of the RULE NAME, TEST STREAM NAME, RULESET NAME, ROW, COL, and LEN fields, see the section Using the Rules Function Panels.

   CUT FROM

   Indicates the location of the field that contains the data you want to reuse. You can use a model screen to fill in the location values.

   Note: Data that you cut is not removed from the screen. It is copied so you can paste it into another screen.

   VALUE OR DESCRIPTION

   Type the name of the field to be cut; optional. The field name can be up to three lines long. In this field you can either put an explanation for why this field is being copied or a sample of the data being copied.

   INPUT/OUTPUT FIELD

   Indicates whether this rule applies to an input or an output screen. Specify I for input; O for output.

3. After you complete the screen and press Enter, the system-generated CUT ID for the piece of data you have selected will appear on the panel.

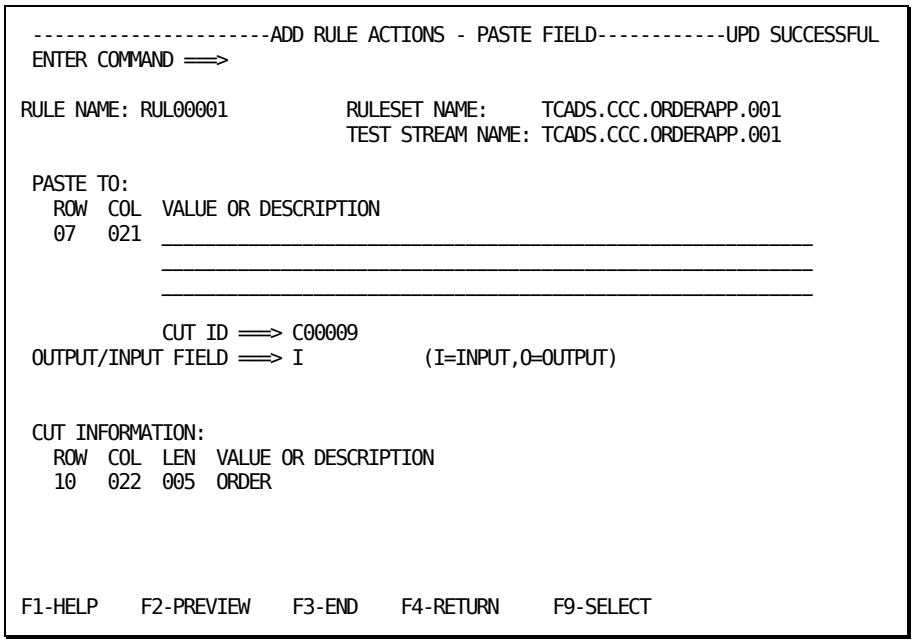
   Note: The CUT ID always begins with the letter C and is followed by five numbers. A sample CUT ID would be C00001. You will use this CUT ID whenever you want to paste this particular data into another panel.

Paste Data Into a Screen

Follow these steps:

1. Type **15** on the Add Rules—Rule Actions Menu to display the Add Rule Actions—Paste Field panel.

   A sample panel is shown next.



2. For a description of the RULE NAME, TEST STREAM NAME, RULESET NAME, ROW, COL, and LEN fields, see the section Use the Rules Function Panels.

   PASTE TO

   Indicates the location of the field where you want the data to be put. You can use a model screen to fill in the location values.

   VALUE OR DESCRIPTION

   Type the name of the field where the data is to be pasted; optional. The field name can be up to three lines long. In this field you can either put an explanation for the paste operation, or a sample of the data being pasted.

   CUT ID

   Type the CUT ID of the data you want to paste into this panel. This ID is generated on the Add Rule Actions—Cut Field panel.

INPUT/OUTPUT FIELD

Indicates whether this rule applies to an input or an output screen. Specify I for input; O for output.

CUT INFORMATION

Displays the location of the field containing the data associated with the CUT ID. Use this information to make sure you are pasting the correct data.

Maintain a Ruleset

Maintaining a ruleset includes:

- Editing a ruleset
- Copying a ruleset
- Deleting a ruleset
- Renaming a ruleset
- Printing a ruleset, and
- Any changes to the rules and rule actions that make up the ruleset

The following sections contain detailed instructions for performing each of these tasks.

Edit Rulesets, Rules and Rule Actions

Follow these steps:

1. Type the name of the ruleset you want to edit (if you know it), or leave the name blank (if you do not know it).
2. Type **E** on the Maintain Rules menu and press Enter.
3. Locate and select the ruleset you want to edit.

   If you did not specify a ruleset name, the Ruleset Selection menu is displayed. (This menu is discussed in the section Browse Through Rulesets and Rules.)

If you specified a ruleset name, the Rules—Summary (Edit) panel for it is displayed. A sample panel is shown next.

```
 ---------------------- EDIT RULES - SUMMARY  ----------------12:53:26
 ENTER COMMAND ===>

RULE NAME: RUL00001            RULESET NAME:     TCADS.CCC.ORDERAPP.001
 LINE   1 TO  14 OF  25       TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 S-EDIT   I-INSERT   D-DELETE   R-REPLICATE    P-PREVIEW
   OBJECT      TYPE ROW  COL LEN OP  VALUE FROM MODEL SCREEN/DESCRIPTION
__ RULESET     T/S                  TCADS.CCCORDER.REL63.001
__  RULE                             RUL00001
__   SCR-RECOG        2   29  23 EQ  CAROL'S COOKIES COMPANY_____________
__    DATAGEN  OUT    1   71  10     01/23/1998+4(d)_____________________
__    DATAGEN  OUT   22   43   8     04/22/97+279(D)_____________________
__    VARIABLE OUT   22   53   1     1___________________________________
__    VARIABLE OUT    2   71   8     08:22:49____________________________
__    CURSOR   OUT    1   33  11     NEW CURSOR ROW-6,COL-17_____________
__    DATAGEN  IN    17   33  11     INFORMATION + 37      _____________
__  RULE                             CUSTINFO
__   VARIABLE  OUT    1    1   1     ____________________________________
__   DELETED   OUT    1    1   1     ____________________________________
__   MOVED     OUT    8    8  10     BIRTHDATE:
__   CHANGED   OUT   15   20   8     TEXT

  F1-HELP  FF3-END  F4-RETURN  F7-UP  F8-DOWN  F9-MODEL SCREEN
```

4. Perform one of the following actions from the panel:

   S

   Edit a ruleset, rule, or rule action.

   I

   Inserts a rule when specified on a ruleset or rule line; inserts a rule action when specified from a rule action line. All insertions are placed *after* the line on which I is specified.

   D

   Deletes a rule or rule action.

   R

   Replicates a rule or rule action.

   P

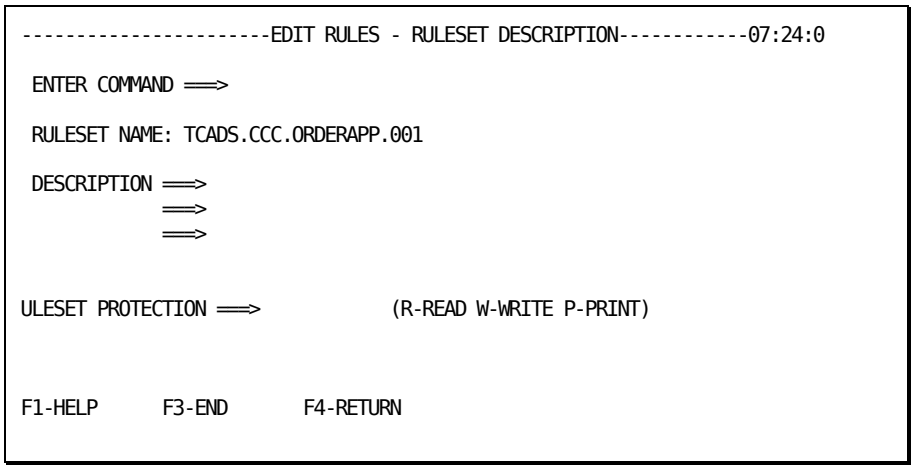
   Previews a rule or rule action.

Edit a Ruleset

Follow these steps:

1. Type an **S** next to RULESET in the OBJECT column and press Enter.

   The Edit Rules—Ruleset Description panel is displayed. A sample panel is shown next.



```
----------------------EDIT RULES - RULESET DESCRIPTION-----------07:24:0

 ENTER COMMAND ===>

 RULESET NAME: TCADS.CCC.ORDERAPP.001

 DESCRIPTION ===>
             ===>
             ===>


ULESET PROTECTION ===>            (R-READ W-WRITE P-PRINT)


F1-HELP      F3-END       F4-RETURN
```

2. On this panel, you can change the ruleset's description and the protection assigned to the ruleset. Press Enter to have your changes accepted, then press PF3 (End) to return to the Summary panel.

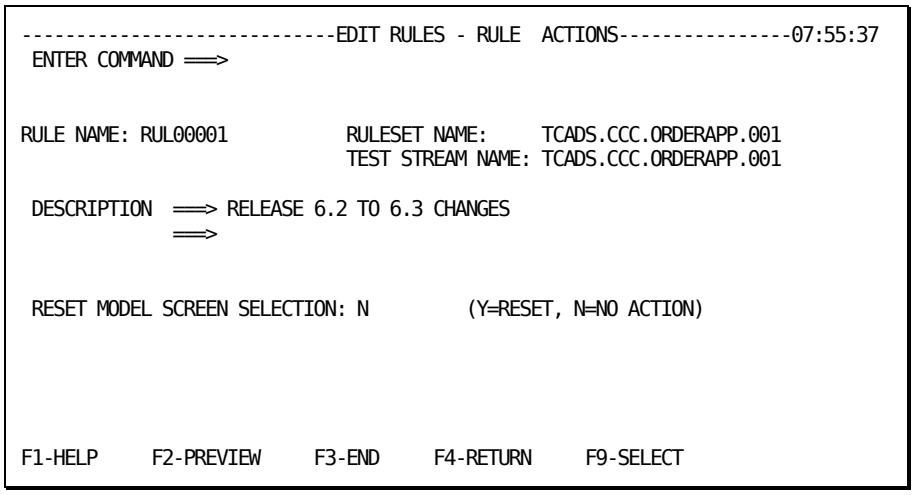
   Note: To change the name of a ruleset, follow the instructions in the section Rename a Ruleset.

Edit a Rule

Follow these steps:

1. Type an **S** next to the rule you want to edit from the Edit Rules—Summary panel, and press Enter.

   The Edit Rules—Rule Actions panel is displayed. A sample panel is shown next.



   On this panel, you can change the rule name and description.

2. Press Enter to have your changes accepted, then press PF3 (End) to return to the Summary panel.

Editing a Rule Action

Follow these steps:

1. Type an **S** to the left of the rule action you want to change.
2. Press Enter to display the completed data entry panel used to define that rule action.
3. Make changes by either overtyping the value or by using the modeling feature (explained in the section Locate and Select the Model Screen*)*.
4. When you are satisfied with your changes, press Enter and then press PF3.

Changing Your Model Screen

Follow these steps:

1. Type Y in the RESET MODEL SCREEN SELECTION field.
2. Press Enter.
3. Press PF9 to select a new model screen.

Copy a Ruleset or Rule

Follow these steps:

1. Type the name of the ruleset you want to copy (if you know it), or leave the name blank (if you do not know it) from the Maintain Rules menu.
2. Type **C** on the Maintain Rules menu and press Enter.

   If you did not specify a ruleset name, the Ruleset Selection menu is displayed. (This menu is discussed in the section Browse through Rulesets and Rules.) Locate and select the rule you want to copy.

   If you specified a ruleset name, the Copy Ruleset panel for it is displayed. A sample panel is shown next.

```
----------------------------COPY RULESET-----------------------13:04:13
ENTER COMMAND ===>

RULESET NAME: TCADS.CCC.ORDERAPP.001

COPY TO:
  DDNAME      ===> TCADS
  APPLICATION ===> CCC
  MEMBER      ===> ORDERAPP
  VERSION     ===> 001

DESCRIPTION ===> SAMPLE RULESET
            ===>
            ===>

COPY RULE:
  RULE NAME  ===>
  NEW NAME   ===>




 F1-HELP      F3-END       F4-RETURN
```

RULESET NAME

Contains the name of the ruleset you are copying.

COPY TO

Specify the destination of the copied ruleset.

DESCRIPTION

Contains the description of the source ruleset. At this time, you can change the description for the new ruleset.

COPY RULE

Use to copy a single rule. Specify the name of the rule you want to copy in RULE NAME. Specify the name of the copied rule in NEW NAME.

3. When you are finished, press Enter to return to your previous screen.

Delete a Ruleset

Follow these steps:

1. Type the name of the ruleset you want to delete (if you know it), or leave the name blank (if you do not know it) from the Maintain Rules menu.
2. Type **D** on the Maintain Rules menu and press Enter.

   If you did not specify a ruleset name, the Ruleset Selection menu is displayed. (This menu is discussed in the section Browse through Rulesets and Rules.) Locate and type **D** next to the ruleset you want to delete. The Confirm Delete of Ruleset panel displays.

   If you specified a ruleset name, the Confirm Delete of Ruleset panel for it is displayed. A sample panel is shown next.

```
----------------------CONFIRM DELETE OF RULESET----------------13:04:28
ENTER COMMAND ===>

RULESET NAME: TCADS.A.A.001

DESCRIPTION:  SAMPLE RULE




STATISTICS:
  CREATION DATE:    09/02/1997
  CREATION TIME:    14:57:32
  CREATED BY:
  LAST UPDATE DATE:
  LAST UPDATE TIME:               TOTAL RULES:        000006
  LAST UPDATED BY:                TOTAL RULE ACTIONS: 000016

TEST STREAM NAME:  TCADS.A.A.001
PROTECTION STATUS:



 F1-HELP      F3-END       F4-RETURN
```

RULESET NAME

Contains the name of the ruleset you are removing.

DESCRIPTION

Contains the description of the ruleset you are removing.

STATISTICS

Displays information relevant to this ruleset.

TEST STREAM NAME

Displays the test stream, if any, that this ruleset is associated with.

PROTECTION STATUS

Displays the protection assigned to this ruleset.

3. When you are finished, press Enter to return to your previous screen.

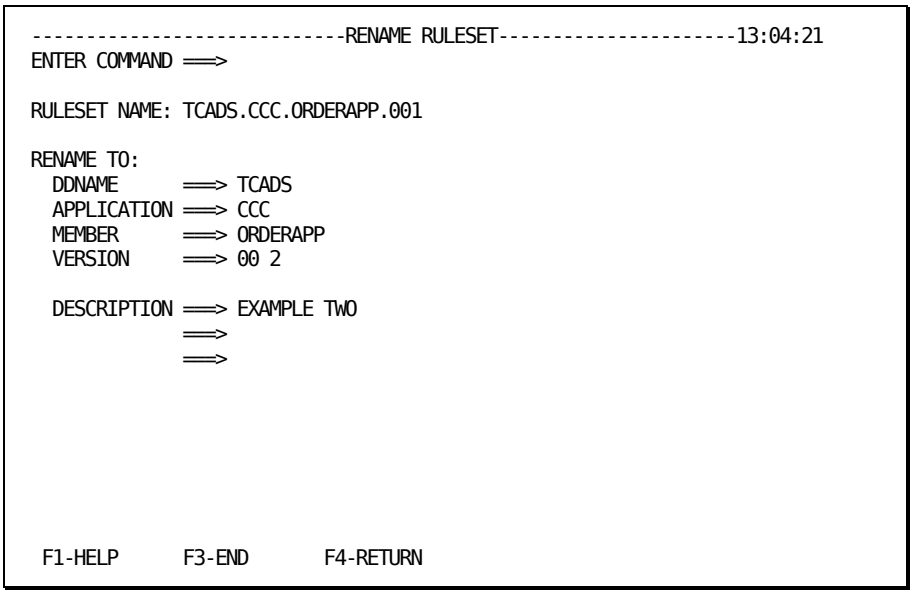
Rename a Ruleset

Follow these steps:

1. Type the name of the ruleset you want to rename (if you know it), or leave the name blank (if you do not know it) from the Maintain Rules menu.
2. Type **R** on the Maintain Rules menu and press Enter.

   If you did not specify a ruleset name, the Ruleset Selection menu is displayed. (This menu is discussed in the section Browse through Rulesets and Rules.) Locate and select the rule you want to rename.

If you specified a ruleset name, the Rename Ruleset panel displays. A sample panel is shown next.



RULESET NAME

Contains the name of the ruleset that you are renaming.

RENAME TO

Specify the new name of this ruleset.

DESCRIPTION

Contains the description from the source ruleset. You can change the description at this time.

3. When you are finished, press Enter to return to your previous screen.

Chapter 9: Inquiry Function

This section contains the following topics:

Overview

Use the Inquiry function to:

- Display a list of all active or suspended functions
- Terminate logging
- Interrupt a run
- Clean up or reconnect CA Verify for CICS sessions

This function is helpful when you want to monitor all CA Verify for CICS activity. The Inquiry function is also the only way you can terminate logging when multiple terminals are being logged, or when the user of a terminal is unaware that his terminal is being logged.

Inquire into Active or Suspended Functions

Follow these steps:

1. Type **I** on the Primary Options Menu.

   CA Verify for CICS displays the Inquiry/Termination menu.

```
  -------------------------  INQUIRY/TERMINATION  -------------10:35:06
ENTER COMMAND ===>                                                  I1

           INVOKED   AT              USING     RECORD
  FUNCTION BY       TERMINAL  TIME   TERMINAL  NUMBER  TEST STREAM NAME
  INQUIRY  CSB      L9D6CA    10:35

_ RUN      ABC      CHDUAL    11:34  VSV1      120     DEMO.DEMOPROG.001
_ LOG      NGN      L8D1F0    11:45  NN01       46     CUST.ORDER.004

  DATA CAPTURE BUFFERS  00% FULL, NO SECONDARY BUFFERS ALLOCATED









TYPE AN "S" NEXT TO YOUR LOG FUNCTION TO STOP IT
TYPE AN "I" NEXT TO A RUN FUNCTION TO INTERRUPT IT (AT ORIGINATING TERMINAL)
    PRESS ENTER TO INVOKE STOP OR INTERRUPT, PRESS PF3 TO END INQUIRY
```

FUNCTION

Identifies the CA Verify for CICS function:

Browse

A test stream is being browsed

Edit

A test stream is being edited

Inquiry

The Inquiry function is being used

Log

A terminal or terminals are being logged

Log Init

A user selected the Log function but logging has not yet begun

Log Full

Logging stopped because the data set is full

Log Term

Logging was either user-terminated or terminated due to an error; if due to an error, the error message number is shown in the RECORD NUMBER column. Refer to the Error Messages section later in this chapter for details.

Option

The Primary Options Menu is being displayed

Rules

The Maintain Rules function is being used

Run

A test stream is being run

Run Init

A user selected the Run function but is not currently running a test stream

Tutorial

The tutorial is being viewed

Utility

One of the utilities is being used

INVOKED BY

The ID of the user who invoked the function.

AT TERMINAL

The terminal at which the function was invoked.

TIME

The time of the last write to the terminal at which the function was invoked. For the Log function, this is the time logging began. If the function was invoked over 24 hours ago, CA Verify for CICS displays the date instead.

USING TERMINAL

For the Log function, the terminal being logged or, for a multiple terminal test stream, the number of terminals being logged.

For the Run function, the virtual terminal being used or, for a multiple terminal test stream, the number of virtual terminals being used.

When terminals are logged under VTAM dynamic install TCTTEs, CA Verify for CICS uses and displays the name of the terminal in use when logging began. The correct terminal will continue to be logged even if the user disconnects from CICS and reconnects onto a different dynamic TCTTE.

RECORD NUMBER

For the Log function, the number of records written to disk

- For a Log Term function due to an error, the error message number appears in this column
- For the Run function, the number of records processed
- For the Browse and Edit functions, the number of the record currently being displayed
- For suspended functions, *SUSP* appears in this column

TEST STREAM NAME

The test stream's name: Application, Member, and Version.

Buffer Utilization

Beneath the list of functions CA Verify for CICS displays this message:

```
DATA CAPTURE BUFFERS  nn% FULL, NO/NUMBER OF SECONDARY
BUFFERS ALLOCATED
```

To avoid affecting system performance, CA Verify for CICS stores screens captured during the Log and Run functions in a buffer before processing them. If the primary buffer becomes full, CA Verify for CICS allocates secondary buffers. This message indicates what percentage of all allocated buffers are full, and how many — if any — secondary buffers have been allocated.

Note: If you find that secondary buffers are frequently allocated, you should increase the size of the primary buffer in the options module. See the *Installation* manual for instructions.

Error Messages

If an error occurs during the Log function, CA Verify for CICS changes the *Log* designation in the Function column to *Log Term* and displays an error message number in the Record Number field. When the log is terminated, CA Verify for CICS displays the error message in the upper right section of the menu. Then you can use the Help (PF1) command to display a longer version of the message.

It's possible, although unlikely, that a note may also be displayed below the buffer utilization message. The following notes may appear:

Buffer Overflow

One or more screens have been lost because the data capture buffer overflowed. Modify the options module to increase the size of the primary buffer or the number of secondary buffers.

Invalid Address

An invalid address was found in a CA Verify for CICS program, control block, or data capture buffer. See the Problem Reporting section of the *Installation* manual.

TCP Exit Abended

The terminal control data capture exit has abended. See the Problem Reporting section of the *Installation* manual.

Terminate Logging

Follow these steps:

1. Type **S** to the left of the Log function and press Enter.

   CA Verify for CICS displays the Log Termination menu. See the "Log Function" chapter for an explanation of this menu.

2. Use this function to terminate:

   - *All* logging, regardless of the Stop Option
   - *Multiple terminal* logging. This is the *only* way logging at multiple terminals can be terminated.
   - *Other terminal* logging when the terminal user is unaware of logging

Interrupt a Run

Follow these steps:

1. Type **I** to the left of the Run function and press Enter.

   CA Verify for CICS displays the Run Status menu with the cancellation message at the terminal where the Run function was invoked.

2. Use this function to cancel long-running test streams with very long cancel intervals.

   Note: To interrupt a run at your terminal, you must invoke the Inquiry function from another terminal.

Clean up and Reconnect Sessions

To clean up a session started at a terminal that is no longer available, type **C** to the left of that function and press Enter. However:

- For Run functions, the CICS task must be terminated before cleanup.
- For Browse functions, cleanup is allowed if the terminal is disconnected; use Reconnect if the terminal is active.
- For Utility functions, cleanup is allowed if the terminal is disconnected; use Reconnect if the terminal is active.
- For Log functions, cleanup is allowed if the terminal is disconnected; use Stop if the terminal is active.

Note: CA Verify for CICS will determine whether cleanup is necessary. If it is not necessary, only the alternate function (Stop or Reconnect) will be allowed.

To reconnect a session left on a terminal, type **R** to the left of that function and press Enter. After you specify Reconnect, pressing Enter at the terminal which owned the session may redisplay the Primary Options Menu.

Reconnect cannot be specified for:

- Run functions, unless the run has not yet begun or has completed
- Transactions, which have been purged or have abended

Use Cleanup instead of Reconnect in these situations.

Terminate the Inquiry

Follow these steps:

1. Use the End (PF3) command.

   CA Verify for CICS redisplays the Primary Options Menu.

Chapter 10: Utilities

This section contains the following topics:

Overview

The utilities make it easy for you to create new test streams to meet your testing needs. For example, you can:

- Copy a test stream and then edit the copy to create a different test case
- Merge terminals and their screens from the same or different test streams into a single test stream to create a large volume test stream for stress or concurrency testing
- Append one test stream onto the end of another test stream or onto itself for stress testing or repetition testing. This utility also lets you copy just the automated variable fields from one test stream to another.
- Insert screens from one test stream into another to modify the test case

The utilities also help you maintain existing test streams. For example, you can:

- Delete test streams you no longer need
- Update a test stream directory to modify its protection status or change the test stream's owner
- Rename a test stream
- Convert a test stream to REXX script
- Secure data in a REXX script

Invoke the Utilities

Follow these steps:

1. Type **U** on the Primary Options Menu.

   CA Verify for CICS displays the following Utilities menu when using the CICS interface.

```
------------------------------  UTILITIES  --------------------------13:50:35
ENTER COMMAND ===>                                                      U1

   C  COPY A TEST STREAM                A  APPEND RECORDS TO A TEST STREAM
   R  RENAME A TEST STREAM              I  INSERT RECORDS INTO A TEST STREAM
   D  DELETE A TEST STREAM              M  MERGE TERMINALS INTO A TEST STREAM
   U  UPDATE A TEST STREAM DIRECTORY

 ENTER TEST STREAM NAME:
   DDNAME      ===> TCADS
   APPLICATION ===> CCC             (LEAVE APPLICATION, MEMBER,
   MEMBER      ===> ORDERAPP         OR VERSION BLANK AND PRESS
   VERSION     ===> 002              ENTER FOR A SELECTION LIST)



 F1-HELP        F3-END        F4-RETURN
```

   When using the TSO interface, CA Verify for CICS displays the following Utilities menu.

```
------------------------------  UTILITIES  --------------------------13:50:35
ENTER COMMAND ===>                                                      U1


   C  COPY A TEST STREAM                A  APPEND RECORDS TO A TEST STREAM
   R  RENAME A TEST STREAM              I  INSERT RECORDS INTO A TEST STREAM
   D  DELETE A TEST STREAM              M  MERGE TERMINALS INTO A TEST STREAM
   U  UPDATE A TEST STREAM DIRECTORY    X  CONVERT A TEST STREAM TO REXX
                                        S  SECURE DATA IN A REXX SCRIPT

 ENTER TEST STREAM NAME:
   DDNAME      ===> TCADS
   APPLICATION ===> CCC             (LEAVE APPLICATION, MEMBER,
   MEMBER      ===> ORDERAPP         OR VERSION BLANK AND PRESS
   VERSION     ===> 002              ENTER FOR A SELECTION LIST)



 F1-HELP        F3-END        F4-RETURN
```

The default ddname and Version values appear; these can be changed.

If you know the test stream you want to process, follow these steps:

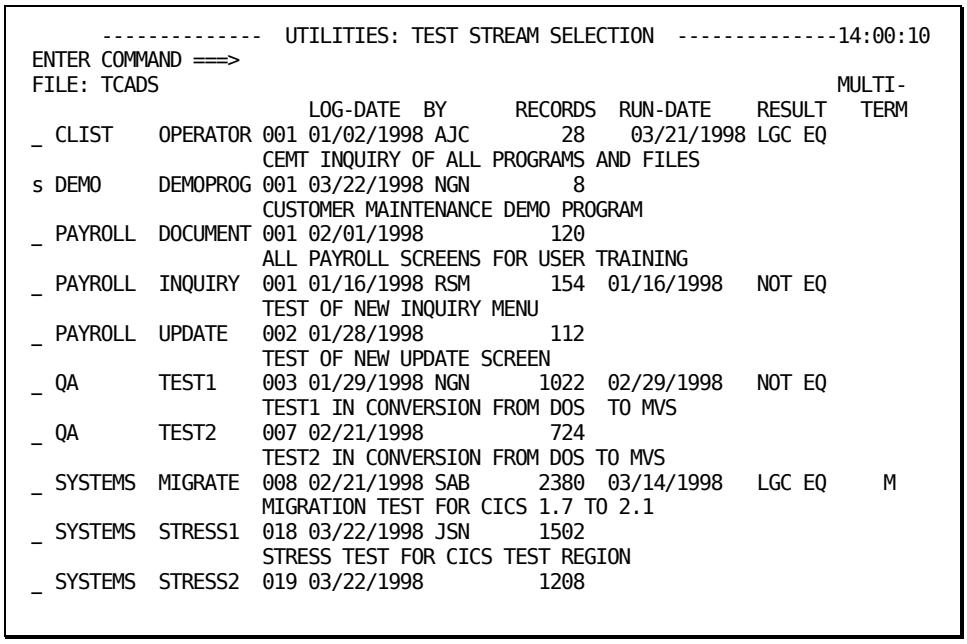
1. Type the Application and Member name and other identifiers, if necessary
2. Type on the command line the letter which identifies the utility you want to use: C, R, D, U, A, I, or M. When using the Utilities menu from the TSO interface, the X and S options are also available.
3. Press Enter.

   CA Verify for CICS displays the menu for the utility you selected.

If you don't know the test stream you want to process, follow these steps:

1. Leave the Application and Member name blank.
2. (Optional) Blank out other identifiers.
3. (Optional) Type on the command line the letter which identifies the utility you want to use: C, R, D, U, A, I, or M. When using the Utilities menu from the TSO interface, the X and S options are also available.
4. Press Enter**.**

   CA Verify for CICS displays the Test Stream Selection menu.



```
     --------------  UTILITIES: TEST STREAM SELECTION  --------------14:00:10
ENTER COMMAND ===>
FILE: TCADS                                                          MULTI-
                        LOG-DATE  BY      RECORDS  RUN-DATE    RESULT   TERM
_ CLIST    OPERATOR 001 01/02/1998 AJC        28    03/21/1998 LGC EQ
                    CEMT INQUIRY OF ALL PROGRAMS AND FILES
s DEMO     DEMOPROG 001 03/22/1998 NGN         8
                    CUSTOMER MAINTENANCE DEMO PROGRAM
_ PAYROLL  DOCUMENT 001 02/01/1998           120
                    ALL PAYROLL SCREENS FOR USER TRAINING
_ PAYROLL  INQUIRY  001 01/16/1998 RSM       154  01/16/1998   NOT EQ
                    TEST OF NEW INQUIRY MENU
_ PAYROLL  UPDATE   002 01/28/1998           112
                    TEST OF NEW UPDATE SCREEN
_ QA       TEST1    003 01/29/1998 NGN      1022  02/29/1998   NOT EQ
                    TEST1 IN CONVERSION FROM DOS  TO MVS
_ QA       TEST2    007 02/21/1998           724
                    TEST2 IN CONVERSION FROM DOS TO MVS
_ SYSTEMS  MIGRATE  008 02/21/1998 SAB      2380  03/14/1998   LGC EQ     M
                    MIGRATION TEST FOR CICS 1.7 TO 2.1
_ SYSTEMS  STRESS1  018 03/22/1998 JSN      1502
                    STRESS TEST FOR CICS TEST REGION
_ SYSTEMS  STRESS2  019 03/22/1998          1208
```

Complete this menu as follows:

- If you typed on the Utilities menu the letter of the utility you want to use, select test streams by typing **S** to the left of each test stream you want to process and press Enter.
- If you did not type on the Utilities menu the letter of the utility you want to use, type the letter — **C, R, D, U, A, I, or M** — to the left of each test stream you want to process and press Enter. When using the Utilities menu from the TSO interface, the **X** and **S** options are also available.

In either case, CA Verify for CICS displays the panel for the utility you selected.

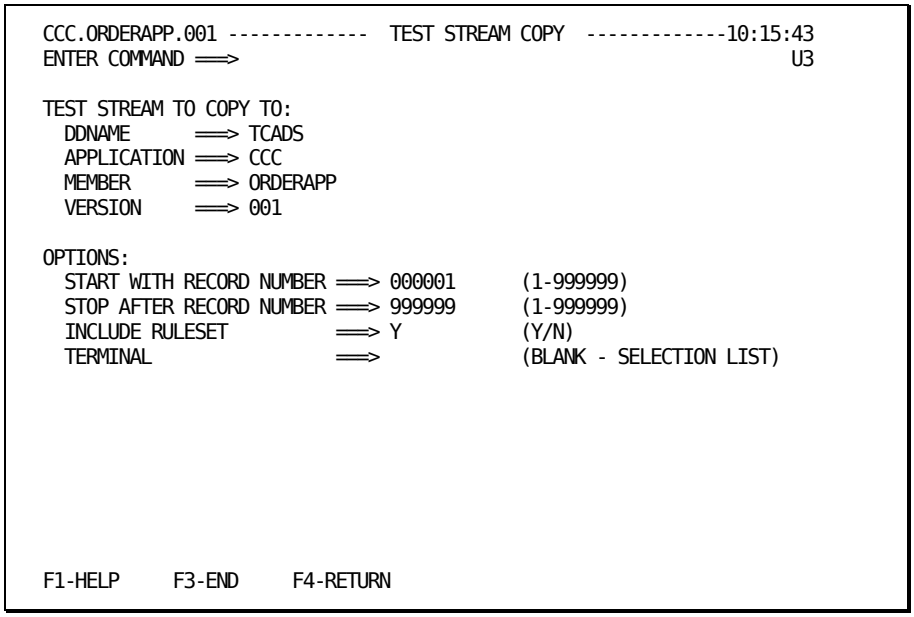
If you select multiple test streams, CA Verify for CICS will redisplay the Test Stream Selection menu each time you press Enter until all selected test streams are processed.

Copy a Test Stream

Follow these steps:

1. Type C on the Utilities or Test Stream Selection menu.

   CA Verify for CICS displays the Test Stream Copy panel.



TEST STREAM TO COPY TO

The name of the test stream being copied appears in the ddname, Application, Member, and Version fields. Modify the name as you wish.

START WITH RECORD NUMBER

The first record to be copied to the output test stream. The default is record 1. Terminal status information is automatically copied because it is required by the Run function.

STOP AFTER RECORD NUMBER

The last record to be copied to the output test stream. The default is record 999999. Use this option to truncate a test stream that ends in the middle of a transaction or at another inconvenient point.

INCLUDE RULESET

Copies the associated ruleset to the corresponding name. Y, the default, copies the ruleset; type N if you do not want the ruleset to be copied.

TERMINAL

Type a terminal name to select one terminal from a multiple terminal test stream. If you leave this field blank, CA Verify for CICS displays the Terminal Selection menu.

```
SAMPLE.MULTTERM.001 ---------  COPY: TERMINAL SELECTION  -------------14:14:40
ENTER COMMAND ===>                                                        UW

SELECT:  "S"    ORIGINAL   TERMINAL     SCREEN      ALTERNATE   VIRTUAL
INQUIRY: "I"    TERMINAL   TYPE         SIZE        SIZE        TERMINAL

         _      A60L2048   3277 REMOTE  24 BY  80   24 BY  80
         _      A60L2049   3277 REMOTE  24 BY  80   24 BY  80
         _      A60L205A   3277 REMOTE  24 BY  80   24 BY  80
         _      A60L205B   3277 REMOTE  24 BY  80   24 BY  80
         ***  END OF TERMINALS  ***









TYPE AN "S" TO SELECT TERMINALS FOR COPY
TYPE AN "I" TO VIEW TERMINAL STATUS
F1-HELP       F3-CONTINUE   F4-RETURN     F7-UP         F8-DOWN
```

This menu lists the following for each terminal in the test stream:

- Original terminal name
- Terminal type; for example, 3277 remote, 3277 local
- Screen size
- Alternate screen size
- Virtual terminal to be used when the test stream is run

2. Type S to the left of one or more terminals and use the Continue (PF3) command. The terminals you select are the ones from which records will be copied.
3. Type I to the left of a terminal for which you want to view Initial Terminal Status information and press Enter. See the "Browse Function" chapter for an explanation of the Initial Terminal Status menu. When you exit from this menu, CA Verify for CICS redisplays the Terminal Selection menu.
4. Use CA Verify for CICS commands as follows:
   - Up (PF7) and Down (PF8) to view additional terminals which don't appear on the initial menu
   - End (PF3) to cancel the copy
5. Press Enter when you have completed the Copy menu.

   CA Verify for CICS copies the test stream and redisplays the Utilities or Test Stream Selection menu, along with a message.
6. Use the End (PF3) command to cancel the copy.

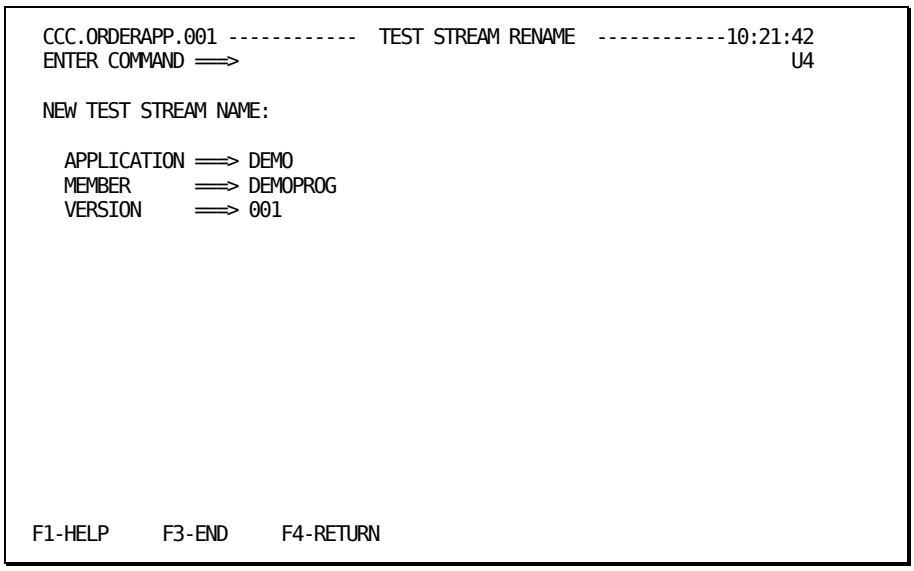
Rename a Test Stream

Follow these steps:

1. Type R on the Utilities or Test Stream Selection menu.

   CA Verify for CICS displays the Test Stream Rename menu.

   Note: If the test stream you are renaming has an associated ruleset, the name of the ruleset will automatically be renamed also.



2. The current test stream name appears in the Application, Member, and Version fields. Modify this name and press Enter.

   CA Verify for CICS renames the test stream and ruleset (if any), and redisplays the Online Utilities or Test Stream Selection menu, along with a message.

3. Use the End (PF3) command to cancel the renaming.

Delete a Test Stream

Follow these steps:

1. Type **D** on the Utilities or Test Stream Selection menu.

   CA Verify for CICS displays the Confirm Delete menu.

   Note: If the test stream you are deleting has an associated ruleset, the ruleset will also be deleted. You will receive a confirmation/warning message before the delete takes place.

```
CCC.ORDERAPP.002 --------------  CONFIRM DELETE  --------------------------
ENTER COMMAND ===>                                                      U5
TCA0548 - CORRESPONDING RULESET WILL ALSO BE DELETED
 DESCRIPTION: CAROL'S COOKIES COMPANY DEMO PROGRAM



            LOG:       RUN:       EDIT:        TEST STREAM:        IN:   OUT:
 INVOKED BY:                                     TOTAL SCREENS:       6     6
 INVOKED ON: 06/02/1998 06/03/1998               AVERAGE BYTES:      46   315
 START TIME: 13:23:04   08:34:50
 DURATION:   00:02:02   00:00:00             LAST RUN:           IN:   OUT:
 SYSTEM:     GRIA5451   GRIA5451               EQUAL:               6     1
 STATUS:     NORMAL     LOGICALLY EQUAL        EQUIVALENT:                5
 TERMINAL:   A60L2048                          IGNORED:                   0
 VSAM CI'S:  2                                 ACCEPTED:

                                               INSERTED:            0

 AVERAGE THINK TIME:    00:00:20.256           DELETED:             0     0
 AVERAGE RESPONSE TIME: 00:00:00.013           CHANGED:             0
 MAXIMUM SCREEN SIZE:   24 BY  80              NOT RUN:             0     0
 PROTECTION STATUS:                                          OWNER:
 ORIGINATING TEST STREAM: CCC.ORDERAPP.001   CREATED BY FUNCTION: RUN

 F1-HELP      F3-CANCEL
```

   This panel lists directory information for the test stream. See the "Browse Function" chapter for an explanation of the directory fields.

2. Press Enter to confirm the deletion; press PF3 to cancel it.

   CA Verify for CICS redisplays the Utilities or Test Stream Selection menu, along with a message.

Update a Test Stream Directory

Follow these steps:

1. Type U on the Utilities or Test Stream Selection menu.

   CA Verify for CICS displays the Directory Update menu.

```
CCC.ORDERAPP.002 -------------  DIRECTORY UPDATE  ---------------------13:51:57
ENTER COMMAND ===>                                                        U6

 DESCRIPTION ===> CAROL'S COOKIES COMPANY DEMO PROGRAM
             ===>
             ===>

             LOG:       RUN:       EDIT:        TEST STREAM:        IN:   OUT:
 INVOKED BY:                                      TOTAL SCREENS:      5      5
 INVOKED ON: 06/29/1998 06/29/1998                AVERAGE BYTES:     52    289
 START TIME: 13:18:52   13:39:44
 DURATION:   00:00:12   00:00:00                 LAST RUN:           IN:   OUT:
 SYSTEM:     GRIA5451   GRIA5451                  EQUAL:              5      1
 STATUS:     NORMAL     NOT EQUAL                 EQUIVALENT:                0
 TERMINAL:   A55TG001                             IGNORED:                   0
 VSAM CI'S:  1                                    ACCEPTED:                  4
                                                  INSERTED:           0      0
 AVERAGE THINK TIME:    00:00:02.388              DELETED:            0      0
 AVERAGE RESPONSE TIME: 00:00:00.002              CHANGED:            0
 MAXIMUM SCREEN SIZE:   24 BY  80                 NOT RUN:            0      0
 PROTECTION STATUS ===>                           OWNER ===>
 ORIGINATING TEST STREAM: CCC.ORDERAPP.001        CREATED BY FUNCTION: RUN

 F1-HELP      F3-END
```

   This menu lists directory information for the test stream. See the "Browse Function" chapter for an explanation of the directory fields.

   The test stream owner — the user who initiated logging — or a security administrator can change these fields:

   - Description
   - Protection Status
   - Owner

   Note: If the test stream is not write-protected, anyone can change the description.

   See the "Log Function" chapter for a discussion of the Description and Protection Status fields.

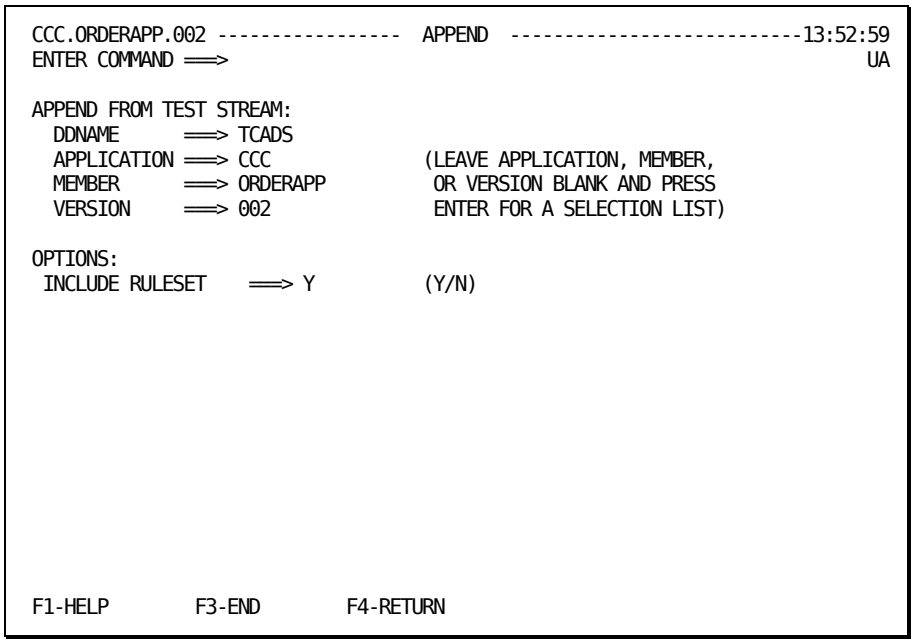
2. Use the End (PF3) command to save the updated directory; use the Cancel command to cancel the update.

Append Records to a Test Stream

Follow these steps:

1. Type A on the Utilities or Test Stream Selection menu.

   CA Verify for CICS displays the Append menu.



   The Append utility copies the test stream you identify on this menu onto the end of the test stream you identified on the Utilities or Test Stream Selection menu.

   Note: Lengthening the test stream may be useful for stress testing. You can also append a test stream to itself for repetition testing. Make sure, however, that the appended records make sense in the context of the records which they follow.

   The following rules apply to the Append utility:

   - Rules will be included in the output ruleset if the Include Ruleset option is Y.
   - When a single terminal test stream is appended to a single terminal test stream, the output is a single terminal test stream.
   - If either of the test streams is a multiple terminal test stream, the output is a multiple terminal test stream. Terminals will be added to the test stream for the terminals in the second test stream which are not in the first.

   APPEND FROM TEST STREAM

   The name of the test stream which is the source of the appended records.

   If you leave the Application, Member, or Version fields blank, CA Verify for CICS displays the Test Stream Selection menu. Type S to the left of the test stream you want to select and press Enter.

INCLUDE RULESET

Leave Y as the default to copy rules when the test stream is appended; type N to exclude them.

The appended fields will be referenced in the next Run, Browse, Edit, or Print function.

2. Press Enter when you have completed the specifications.

   CA Verify for CICS either performs the append or displays the Append Confirmation menu. If CA Verify for CICS performs the append, it redisplays the Utilities or Test Stream Selection menu, along with a message.

Confirm the Append

If the output test stream contains two consecutive input or output screens from the same terminal, CA Verify for CICS displays the Append Confirmation menu. The message on this menu indicates whether CA Verify for CICS found two consecutive input or output screens.

Here's how either situation could occur. Suppose test stream A is being appended to test stream B. If the first screen for test stream A is an input screen and the last screen for test stream B is an input screen, the output test stream would contain two consecutive input screens. In this case, you may want to edit test stream B to delete the last input screen.

Similarly, if the first screen for test stream A is an output screen and the last screen for test stream B is an output screen, the output test stream would contain two consecutive output screens. In this case, you may want to edit test stream A to delete the first output screen.

Press Enter to confirm the Append or use the End (PF3) command to cancel it. If you confirm the Append, CA Verify for CICS appends the records and redisplays the Utilities or Test Stream Selection menu, along with a message. If you cancel the Append, CA Verify for CICS redisplays the Append menu.

Insert Records into a Test Stream

Follow these steps:

1. Type I on the Utilities or Test Stream Selection menu.

   CA Verify for CICS displays the Insert menu.

   Note: If the records you are inserting use a different ruleset than the test stream you are inserting them into, then the rules from both test streams will be combined.

```
CCC.ORDERAPP.001 ----------------  INSERT  --------------------10:24:22
ENTER COMMAND ===>                                                UI


INSERT FROM TEST STREAM:
  DDNAME      ===> TCADS
  APPLICATION ===> CCC                        (LEAVE APPLICATION, MEMBER
  MEMBER      ===> ORDERAPP                    OR VERSION BLANK AND PRESS
  VERSION     ===> 001                         ENTER FOR A SELECTION LIST)


OPTIONS:
  START WITH RECORD NUMBER      ===> 000001     (1-999999)
  STOP AFTER RECORD NUMBER      ===> 999999     (1-999999)
  INSERT AFTER RECORD NUMBER    ===> 999999     (1-999999)
  INCLUDE RULESET               ===> Y          (Y/N)


MULTIPLE TERMINAL TEST STREAM OPTIONS:
  TERMINAL TO SELECT FROM    ===>
  TERMINAL TO INSERT AS      ===>




F1-HELP     F3-END     F4-RETURN
```

The Insert utility inserts records from one or more test streams into the test stream you identified on the Utilities or Test Stream Selection menu. You can also insert records from a test stream into that same test stream. Make sure, however, that:

- The records from both test streams are from similar types of terminals
- The inserted transactions will work where they are inserted

Inserting records is one way of expanding a test stream and changing the testing scenario.

INSERT FROM TEST STREAM

The name of the test stream which is the source of the inserted records.

If you leave the Application, Member, or Version fields blank, CA Verify for CICS displays the Test Stream Selection menu. Type S to the left of the test stream you want to select and press Enter.

START WITH RECORD NUMBER

The first record to be inserted. The default is record 1.

STOP AFTER RECORD NUMBER

The last record to be inserted. The default is record 999999; such as, the last record in the test stream.

INSERT AFTER RECORD NUMBER

The record in the output test stream after which the records should be inserted.

- If you specify 0, the records are inserted at the beginning of the test stream.
- If you leave 999999 as the default or specify a number greater than the number of records in the test stream, the records are inserted at the end of the test stream.

INCLUDE RULESET

Indicates whether the ruleset from the target test stream should be inserted into the source test stream. Y, the default, inserts the ruleset; N does not.

The last two options apply to multiple test streams only.

TERMINAL TO SELECT FROM

The name of the terminal in the source test stream from which records are to be selected. If you leave this field blank, CA Verify for CICS displays the From Terminal Selection menu.

TERMINAL TO INSERT AS

The terminal name that should be used for the inserted records in the output test stream. If you leave this field blank, CA Verify for CICS displays the To Terminal Selection menu.

2. Press Enter when you have completed the specifications.
   - If necessary, CA Verify for CICS displays the From Terminal Selection and/or To Terminal Selection menus. See the Select a Terminal section next.
   - If necessary, CA Verify for CICS displays the Insert Confirmation menu. See the Confirm the Insert section next.
   - Otherwise, CA Verify for CICS displays the Insert Status menu. *See the* Reviewing the Insert Status section next.

Select a Terminal

For multiple terminal test streams, CA Verify for CICS displays the From Terminal Selection menu if you leave the Terminal to Select from field blank, and the To Terminal Selection menu if you leave the Terminal to Insert As field blank.

Follow these steps:

1. Type S to the left of *one* terminal and use the End (PF3) command.
   - The terminal you select on the From Terminal Selection menu is the terminal from which records will be selected for insertion.
   - The terminal you select on the To Terminal Selection menu is the terminal which will be associated with the inserted records.
2. Type I to the left of a terminal for which you want to view Initial Terminal Status information and press Enter. See the "Browse Function" chapter for an explanation of the Initial Terminal Status menu. When you exit from this menu, CA Verify for CICS redisplays the From or To Terminal Selection menu.
3. Use CA Verify for CICS commands as follows:
   - Up (PF7) and Down (PF8) to view additional terminals which don't appear on the initial menu
   - Cancel to cancel the insertion

Confirm the Insert

If the output test stream contains two consecutive input or output screens from the same terminal, CA Verify for CICS displays the Insert Confirmation menu. The message on this menu indicates whether CA Verify for CICS found two consecutive input or output screens.

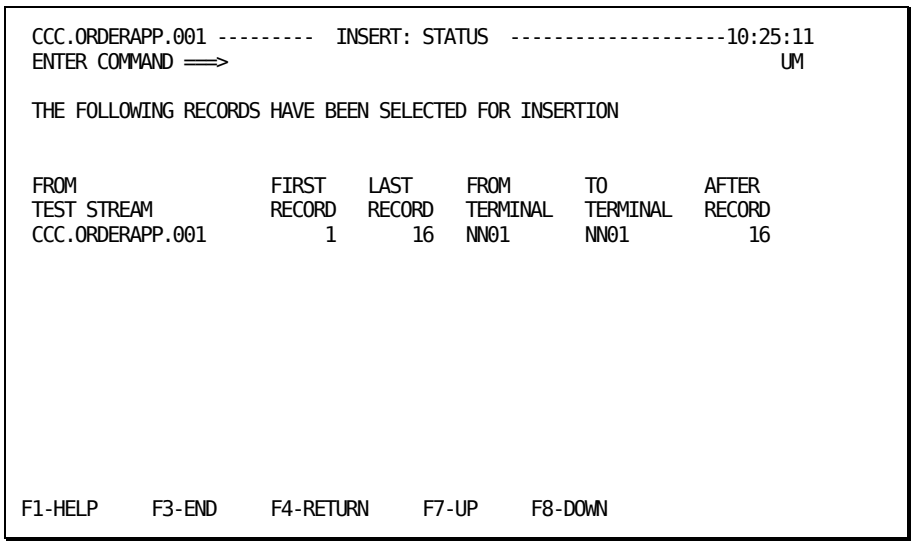
If this situation is acceptable — for example, your application issues consecutive reads or writes — press Enter to confirm the Insert. CA Verify for CICS displays the Insert Status menu. Otherwise, use the End (PF3) command to cancel it. CA Verify for CICS then redisplays the Insert menu so you can modify the record numbers.

Review the Insert Status

Follow these steps:

1. Specify all the information required for record insertion.

   CA Verify for CICS displays the Insert Status menu.



```
CCC.ORDERAPP.001 ---------  INSERT: STATUS  --------------------10:25:11
ENTER COMMAND ===>                                                 UM

THE FOLLOWING RECORDS HAVE BEEN SELECTED FOR INSERTION


FROM                 FIRST    LAST     FROM       TO         AFTER
TEST STREAM          RECORD   RECORD   TERMINAL   TERMINAL   RECORD
CCC.ORDERAPP.001          1       16   NN01       NN01           16









F1-HELP     F3-END     F4-RETURN     F7-UP     F8-DOWN
```

2. If your specifications are correct, use the Enter key to perform the insertion and save the output test stream.

   CA Verify for CICS inserts the records and redisplays the Utilities or Test Stream Selection menu, along with a message.

3. If your specifications are incorrect, use the End (PF3) key to cancel the insertion.

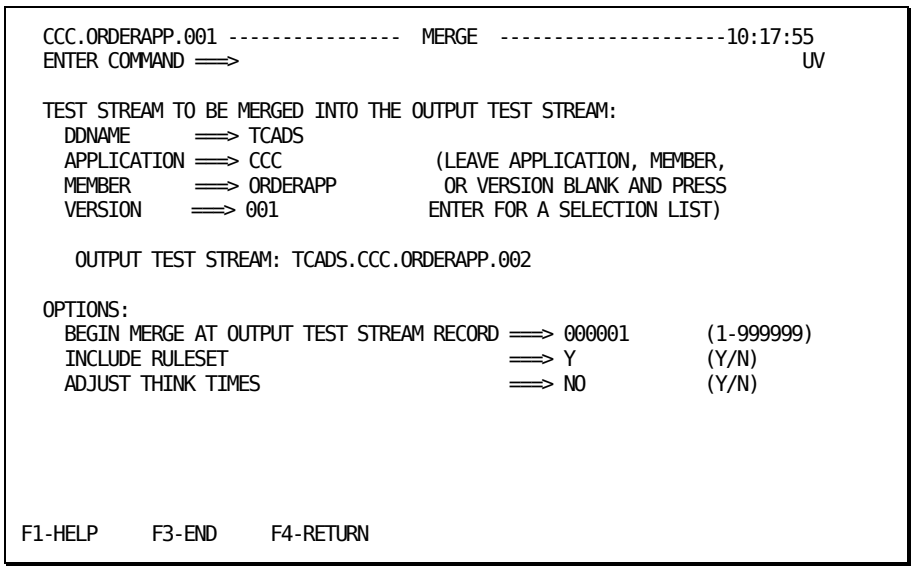
Merge Terminals into a Test Stream

Follow these steps:

1. Type M on the Utilities or Test Stream Selection menu.

   CA Verify for CICS displays the Merge menu.

   Note: If the test streams being merged have associated rulesets, then the rulesets are also merged.



   The Merge utility merges terminals and their records from a test stream into the test stream you identified on the Utilities or Test Stream Selection menu. You can merge records from a test stream into the same test stream or into a different test stream.

   The new records are interspersed among the existing records in the test stream. CA Verify for CICS calculates the position for each record to be merged as follows:

   - First, CA Verify for CICS calculates its offset from the beginning of the test stream (think times plus response times for the terminal).
   - Then, CA Verify for CICS inserts the record at the corresponding offset in the test stream into which it is being merged.

   You can modify the record's placement by specifying a record number after which the new records should be interspersed. You can also adjust the think times of the merged records.

Merging records is an excellent way of creating different testing situations. For example, you can merge a test stream with itself for concurrency testing. Or, you can merge several test streams to create a large test stream for stress testing.

TEST STREAM TO BE MERGED INTO THE OUTPUT TEST STREAM

The name of the test stream which is the source of the merged records.

If you leave the Application, Member, or Version fields blank, CA Verify for CICS displays the Test Stream Selection menu. Type S to the left of the test stream you want to select and press Enter.

OUTPUT TEST STREAM:

Identifies the name of the output test stream. This is the name you specified on the Utilities or Test Stream Selection menu.

BEGIN MERGE AT OUTPUT TEST STREAM RECORD

The record in the output test stream after which the records should be merged. The default is the first record. Specify a record number to indicate the point in the test stream at which CA Verify for CICS should begin interspersing the merged records.

INCLUDE RULESET

Indicates whether the ruleset from the source test stream should be merged into the target test stream. Y, the default, merges the ruleset; N does not.

ADJUST THINK TIMES

Type Yes if the two test streams are of different durations and you want them to be the same. CA Verify for CICS will adjust the think times of the test stream to be merged so it is approximately as long as the test stream into which it is being merged.

2. Press Enter when you have completed the specifications.
   - If necessary, CA Verify for CICS displays the Terminal Selection menu. See the Select Terminals section next.
   - Otherwise, CA Verify for CICS displays the Merge Status menu. See the Review the Merge Status section next.

Select Terminals

If the test stream to be merged is a multiple terminal test stream, CA Verify for CICS displays the Terminal Selection menu.

Follow these steps:

1. Type S to the left of each terminal you want to select and use the End (PF3) command.

   Note: You must select at least one terminal.

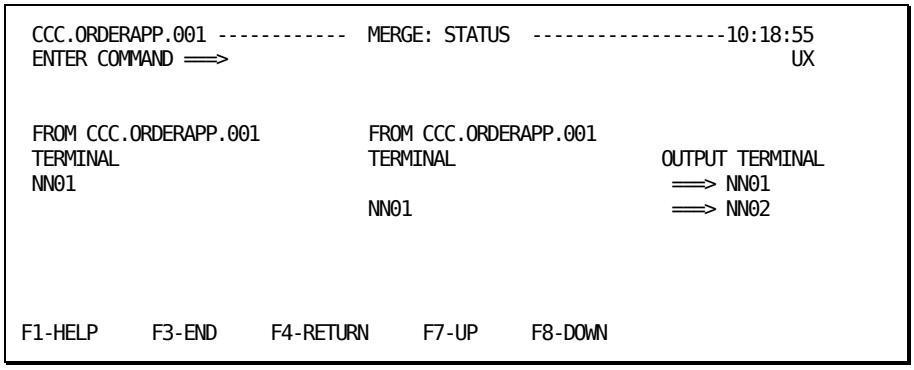
2. Type I to the left of each terminal for which you want to view Initial Terminal Status information and press Enter. See the "Browse Function" chapter for an explanation of the Initial Terminal Status menu. When you exit from the Initial Terminal Status menu, CA Verify for CICS redisplays the Terminal Selection menu.
3. Use CA Verify for CICS commands as follows:
   - Assign to insert S to the left of every terminal
   - Reset to insert _ (the initial setting) to the left of every terminal
   - Up (PF7) and Down (PF8) to view additional terminals which don't appear on the initial menu
   - Cancel to cancel the merge

Review the Merge Status

Follow these steps:

1. Specify the Merge information.

   CA Verify for CICS displays the Merge Status menu.



   - The first From column identifies terminals from the "to" test steam.
   - The second *From* column identifies terminals from the source test stream.
   - The *Output Terminal* column names the output terminals.
2. If the specifications are correct, use the End (PF3) command to perform the merge and save the output test stream.

   CA Verify for CICS merges the test streams and redisplays the Utilities or Test Stream Selection menu, along with a message.

   If any of the terminals in the test stream to be merged have the same name as a terminal in the test stream into which they are being merged, CA Verify for CICS assigns a new name to be used for that terminal in the output test stream. The new name appears in the Output Terminal column.

You can change the name of any terminal listed on the Merge Status menu; however, *each name must be unique*.

3. Use CA Verify for CICS commands as follows:
   - Up (PF7) and Down (PF8) to scroll through the terminal list
   - Cancel to cancel the merge

Merge Considerations

When CA Verify for CICS runs multiple terminal test streams, it recreates, as much as possible, the same concurrences and order of events that existed when the test stream was originally logged.

During logging, CA Verify for CICS records input and output screens in the order in which they pass through terminal control. During a run, input screens are transmitted in the same order in which they were logged. The application controls the order of outputs, but each time CA Verify for CICS finds an output screen in the test stream, it waits for the application to write an output screen before transmitting subsequent input screens.

This method ensures that multiple terminal test streams produce consistent results. When you merge test streams, you must make sure that the new order of events represents a possible one. Failure to do so may cause mismatches, missing output, or a *deadly embrace* situation when you run the new test stream.

Convert a Test Stream to a REXX Script

Follow these steps:

1. While using the CA Verify for CICS TSO interface, type X on the Utilities or Test Stream Selection menu.

   CA Verify for CICS displays the Convert Test Stream to REXX menu.

```
ASMDEMO.OPTION1.001 --------------  CONVERT TO REXX  ------------------17:14:50
ENTER COMMAND===>                                                        UE

TEST STREAM TO CONVERT: TCADS.ASMDEMO.OPTION1.001

REXX SCRIPT FILE:
  DATA SET NAME   ===>'USER02.VERIFY.SCRIPT'
  MEMBER NAME     ===>        (BLANK OR PATTERN FOR MEMBER SELECTION LIST)

OPTIONS:
  APPLID          ===>        (LOGON APPLID)
  SCREEN LINES    ===>03      (# OF SCREEN LINES TO INCLUDE: 0-43)
  PROTECT TARGET  ===>N       (Y/N)
  SHARE PDS       ===>N       (Y/N)
  DELAYS          ===>Y       (Y/N)

ACTIVE PASSWORD SECURITY:
  CAPTURE FORMAT . . .TOKEN       USERID . . . . . .RYARO02




F1-HELP       F3-END       F4-RETURN
```

TEST STREAM TO CONVERT

Identifies the name of the test stream to be converted. This is the name you specified on the Utilities or Test Stream Selection menu.

DATA SET NAME

The data set name of the file in which to store the converted REXX script. The data set may be sequential or partitioned (PDS or PDSE) and must be catalogued, with a fixed record format and an 80-byte record length. The default value is userid.VERIFY.SCRIPT. Modify the name as you wish.

MEMBER NAME

The name of the REXX script member, if the data set is partitioned (PDS or PDSE). It is blank by default. Specify the name of the REXX script to which the test stream will be converted, or leave the field blank for a member selection list.

APPLID

The logon APPLID for the CICS region on which the converted REXX script will be executed. It is blank by default. If no APPLID is specified, the LOGON statement is generated as "LOGON '????????'" in the REXX script. The question marks will need to be replaced with a valid APPLID before the script can be executed.

SCREEN LINES

The number of lines from each screen image to be included as comments in the REXX script. The default value is 3. Enter a value from 0 to 43.

PROTECT TARGET

Indicates whether a REXX script should be protected if it already exists in the script file. The default value is 'N'. Enter 'Y' to prevent an existing REXX script from being overlaid.

SHARE PDS

Indicates whether multiple jobs and users can write to the same script file at the same time, if the script file is a PDS. The default value is 'N'. Enter 'Y' to share the script file PDS.

DELAYS

Indicates whether "DELAY" statements, which represent "user think time" should be included in the converted REXX script. The default value is 'Y'. Enter 'N' to omit "DELAY" statements.

CAPTURE FORMAT

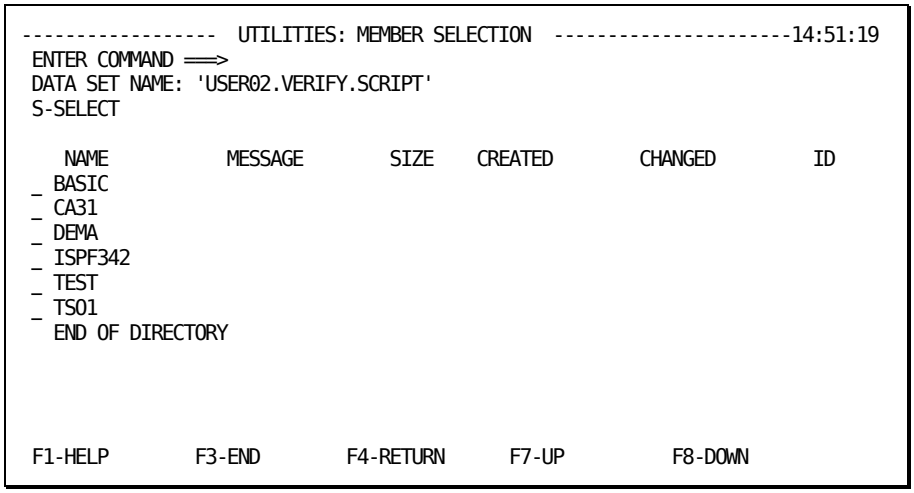
The active password capture format used when converting a test stream to a REXX script. The password capture format may be modified from the REXX Password Security panel, or by updating and assembling default options member, VTEOT.

USERID

The active userid. This is a display field only and cannot be modified.

2. If the REXX script file is a PDS and you leave the member name blank,

   CA Verify for CICS displays the Member Selection menu.



Select a member by typing S to the left of the member name and press Enter

Note: When converting a test stream to REXX the goal is to create a script that can be executed without changes. However the script may not exactly match everything that is in a test stream.

CA Verify for CICS logging normally starts at a clear screen and when executed, they will be presented with the site's "Welcome to CICS" screen. To resolve this, conversion will create a TYPE CLEAR statement that is not in the original test stream.

INVITE statements are required when an application unlocks the keyboard, but may subsequently send more output before input keystrokes are expected. Technically this is a between bracket issue. For example, this sometimes occurs during a TSO logon. The conversion creates INVITE statements as required, based on the converted test stream. However, since the number of outputs sent in this condition may vary, the script may have too many or too few INVITE statements.

An INVITE functions as a WAIT, and may slow the execution down if too many are coded.

Secure Data in a REXX Script

If passwords were captured in REXX scripts converted from test streams or if passwords were coded in user written scripts or execs, then the REXX script would contain visible passwords that could be used to gain unauthorized access to protected system resources.

CA Verify for CICS provides two password security options that allow passwords to be captured and used by CA Verify for CICS REXX scripts while not being disclosed to or used by anyone other than their owners. There is also a third option, in which actual passwords are captured and are visible in converted REXX scripts. This option should only be used for test systems that do not need to be secure.

Token Security

Token Security captures symbolic password tokens instead of passwords in REXX scripts converted from test streams and allows the scripts to enter tokens instead of passwords to sign on.

Code Security

Code Security captures encoded password codes instead of passwords in REXX scripts converted from test streams and allows the scripts to enter codes instead of passwords to sign on.

Password Security

Password Security captures actual passwords in REXX scripts converted from test streams. This means that the passwords are visible in the scripts. This option should only be used on test systems which do not need to be secure.

The following topics discuss these security measures.

Token Security

Token Security is a password encryption system that allows CA Verify for CICS test streams that have been converted to REXX scripts to use passwords. A token protects test streams from being disclosed to or used by anyone other than their owner.

Token Security works with RACF, ACF2, CA Top Secret or any other security system that reliably establishes the user ID that owns address spaces in which CA Verify for CICS is used in the manner prescribed by the z/OS System Authorization Facility (SAF).

- A password is any value entered in a non-display screen field.
- A token is a name that may be used in CA Verify for CICS REXX scripts to enter a user's password.

Token Security may be used to capture tokens instead of passwords when a CA Verify for CICS test stream is converted to REXX.

When a CA Verify for CICS test stream is converted to REXX using Token Security and a password is encountered, the password is replaced by a token in the REXX script. The token and encrypted passwords are stored in the user's profile. The token is either an existing token for the password found in the user's profile or a new token that is automatically created for the password.

When a new token is created for a user:

- A unique name is selected to use as the new token.
- A profile is created for the user, if needed, to store private data.
- The password is combined with the user ID that owns the address space.
- The combined password and user ID is encrypted.
- The new token and the encrypted data are stored in the user's profile.
- A token library is created for the user, if needed, to store token screens.
- An image of the screen that contains the password field is stored as a token screen.
- If the password was entered by a person and system options specify password capture notification, the user is notified that a password was captured.

A TOKEN footnote identifies each token captured in a converted screen image. The footnote in the following example identifies a captured token, UID1A, at screen row 8, column 20.

```
+Session Step(VTERM001 ENTER--> A31ITSO) Cursor(8,20) ------
|.------------------------------ TSO/E LOGON ----------------------------------
|.
|.
|   .Enter LOGON parameters below:                                          :
|
|  . Userid    ===>.BUCBR01.                       .
|
|  . Password  ===>.USER02A.                      .
|
+- TOKEN AT <8 20> ------------------------------------------------------------
```

The CA Verify for CICS TYPE command TOKEN keyword identifies a token in a converted or user-written REXX script. For example:

```
" TYPE <8,20> TOKEN 'UID1C' ",
" <8,24> ENTER "
```

When a CA Verify for CICS REXX script is used to create a session and a token is used to enter a password, the user ID that owns the address space is used to decrypt the token data retrieved from the user's profile. The correct password can be recovered from the token data only in an address space owned by the user id that was used to encrypt the password.

Token Security not only allows CA Verify for CICS REXX scripts to enter correct passwords that are not seen in the scripts, but also ensures that passwords represented by tokens are used only by their owners.

Token Security provides other safeguards that protect passwords from accidental disclosure, protect passwords sent by online systems to terminals, and protect passwords while they are in memory.

- To prevent accidental disclosure of passwords that could occur if tokens were incorrectly typed in display fields, the CA Verify for CICS TYPE command does not allow tokens to be typed into display fields.

- Some widely-used security systems and session managers rely on 3270 non-display fields to hide displayed passwords and use them not only as password input fields, but also use both modifiable and protected non-display fields for other purposes.

Examples:

- A session manager that supports concurrent virtual sessions with multiple online systems remembers passwords when users sign on to the session manager. Subsequently, when a user selects a system from a session manager menu, the session manager starts a virtual session with the system. When the selected system sends a sign on display to the virtual terminal, the session manager fills in the non-display password input field for the user before sending the sign on display to the real terminal, allowing the user to simply press the enter key to sign on to selected systems.
- One security system is known to save passwords in protected non-display fields. When a user enters a current password and a new password to change passwords, the system sends a verify-password display that saves the user's password in a protected non-display field. When the new password is re-entered by the user, both the saved password and the re-entered password are sent to the system. Like many security systems, the system views all passwords as uppercase characters, and the password saved in the protected non-display field reflects its uppercase standard, even though it was entered by the user as lowercase characters.
- When a user enters a password to sign on, another security system responds by sending only a protected non-display attribute that replaces the modifiable non-display attribute of the password input field and leaves the password in the display.

When CA Verify for CICS test streams are converted to REXX, screen images are captured that show each request sent by a user to a system and each response sent by the system to the terminal. To protect all passwords in captured data regardless of who sent them, Token Security replaces all data in modifiable non-display fields with tokens, and replaces data in protected non-display fields with tokens, if the data or a lower case version of the data matches a password referred to by any existing user token.

View and Modify a Token

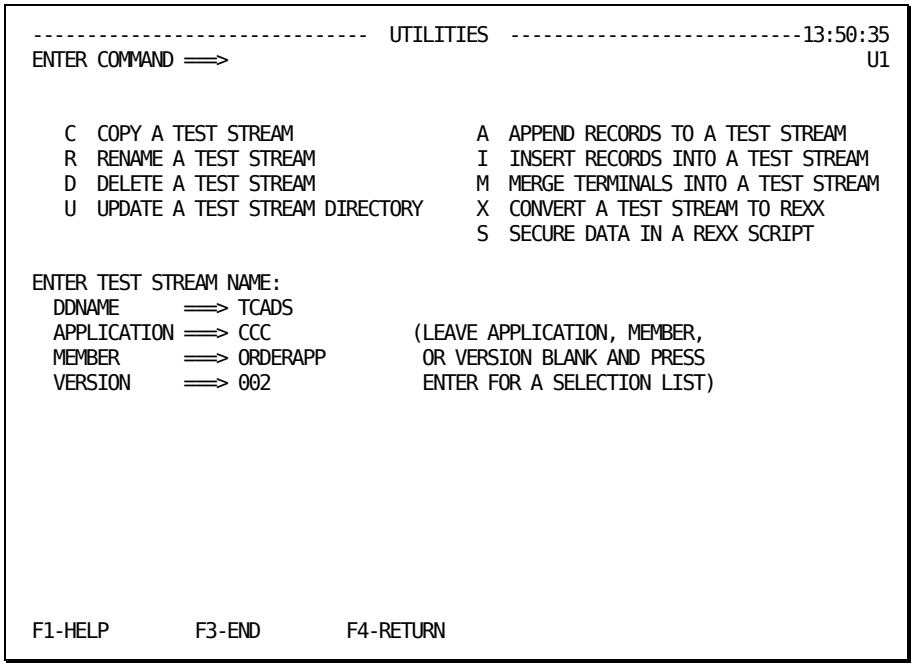
If a password is changed, and token security is used, then the token value (password) must be changed to the new password using the Token Security Panel. You can view and modify tokens from the Utilities menu of the TSO interface.

Note: The Token Security panel is not accessible through the CICS interface.

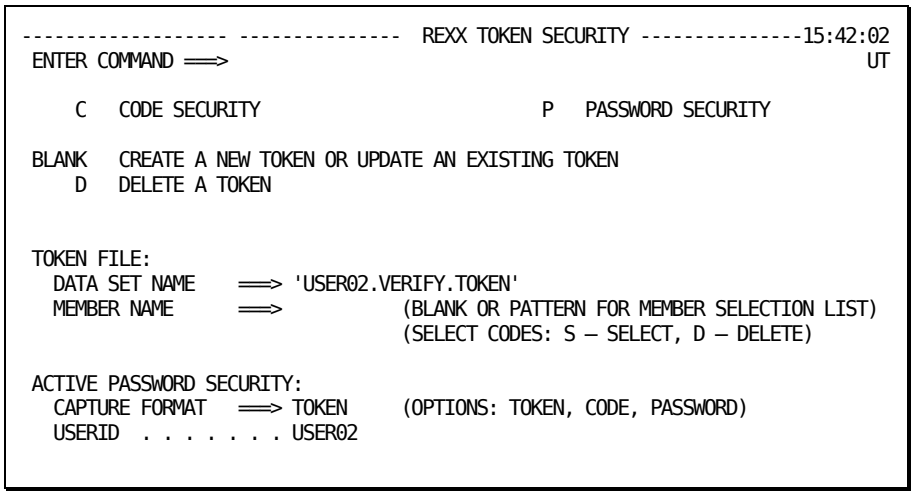
Follow these steps:

1. Type U on the Primary Options Menu of the CA Verify for CICS TSO interface.

   CA Verify for CICS displays the Utilities menu.



2. Select option S. The Token Security panel displays if token security is active for the user. If the token security is not active, display the token security panel by typing '**T**' on the command line of the displayed security panel and hit Enter. The token security panel follows:



DATA SET NAME

Specify the data set name of the token file. The data set must be partitioned (PDS or PDSE), with a fixed record format and a 135-byte record length. The default value is userid.VERIFY.TOKEN.

MEMBER NAME

Specify the name of the REXX token member. It is blank by default. Specify a member name or leave the member name blank for a member selection list.

CAPTURE FORMAT

Displays the active password capture format used when converting a test stream to a REXX script. If options are listed following the Capture Format field, a listed option may be entered to change the active capture format.

The capture format controls password security for all CA Verify REXX scripts for the user, regardless of when the scripts were created.

USERID

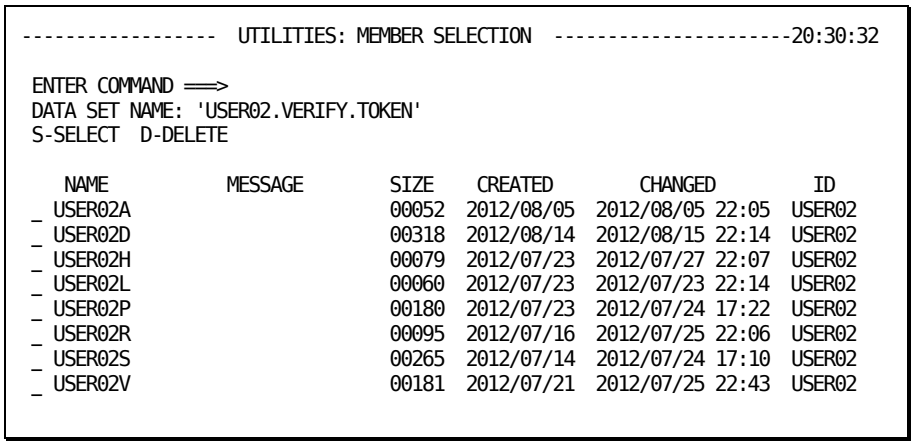
The active userid. This is a display field only and cannot be modified.

You can perform the following actions on this screen.

- To update a token specify the name of the member in the token file and hit Enter. Alternately, leave the member name blank, or specify a pattern to display a member list.
- To create a new token from an existing token, specify the dataset name and member name of the existing token and hit Enter. The member name may be left blank, or a pattern may be specified, to display a member list.
- To delete a token enter a 'D' on the command line and specify the name of the token file member, then hit Enter. The member name may be left blank, or a pattern may be specified, to display a member list.

- To change the active capture format for the userid, specify one of the listed capture format options and hit Enter. The new capture format is stored in the user's profile and the panel for the specified capture format is displayed.
- To display the code or password security panel without changing the active capture format, type 'C' or 'P' on the command line and hit Enter.

  Note: The code security and password security options are displayed below the command line if they are available. The options are available if the settings in the REXX options table allow them.
- If the member name is left blank, or a pattern is specified, the following Member Selection List is displayed:



```
------------------  UTILITIES: MEMBER SELECTION  ----------------------20:30:32

 ENTER COMMAND ===>
 DATA SET NAME: 'USER02.VERIFY.TOKEN'
 S-SELECT  D-DELETE

    NAME           MESSAGE        SIZE    CREATED        CHANGED          ID
 _ USER02A                       00052  2012/08/05  2012/08/05 22:05  USER02
 _ USER02D                       00318  2012/08/14  2012/08/15 22:14  USER02
 _ USER02H                       00079  2012/07/23  2012/07/27 22:07  USER02
 _ USER02L                       00060  2012/07/23  2012/07/23 22:14  USER02
 _ USER02P                       00180  2012/07/23  2012/07/24 17:22  USER02
 _ USER02R                       00095  2012/07/16  2012/07/25 22:06  USER02
 _ USER02S                       00265  2012/07/14  2012/07/24 17:10  USER02
 _ USER02V                       00181  2012/07/21  2012/07/25 22:43  USER02
```

- To display the original input screen for which the token was created, select a member and hit Enter.
- To delete a token, type '**D**' in the field to the left of the member name and hit Enter.

Example: Screen for a Token for a Script which logs on to a TSO Session

If the token was created for a script which logs on to a TSO session, the original screen looks like the following:

```
------------------------------ TSO/E LOGON ----------------------------------
TCA0579 - PRESS ENTER TO PROCEED OR PF3 TO CANCEL.

    Enter LOGON parameters below:                  RACF LOGON parameters:

    Userid    ===> USER1

    Password  ===>                                 New Password ===>

    Procedure ===> TSOPROC                         Group Ident  ===>

    Acct Nmbr ===> 123456789

    Size      ===> 4096

    Perform   ===>

    Command   ===>

    Enter an 'S' before each option desired below:
          S -Nomail       S -Nonotice      S -Reconnect        -OIDcard

 PF1/PF13 ==> Help    PF3/PF15 ==> Logoff    PA1 ==> Attention    PA2 ==> Reshow
 You may request specific help information by entering a '?' in any entry field
```

Hit Enter to proceed to the 'Change Token' screen, the 'Create token' screen or the 'Confirm Delete screen.

Example: Change Token screen

```
----------------------------  CHANGE TOKEN  --------------------------16:16:13
 ENTER COMMAND ===>                                                      US

 TOKEN FILE:
   DATA SET NAME  . . .  'USER02.VERIFY.TOKEN'
   MEMBER NAME  . . . .  USER02B


 ENTER A NEW PASSWORD TO CHANGE THE TOKEN.

   NEW PASSWORD     ===>
   CONFIRM PASSWORD ===>
```

To update the token, specify a new password in the non-display ‘New Password’ and 'Confirm Password' fields and hit Enter.

Example: Create Token screen

```
----------------------------  CREATE TOKEN  -------------------------16:16:13
 ENTER COMMAND ===>                                                     US

 TOKEN FILE:
   DATA SET NAME  . . .  'USER01.VERIFY.TOKEN'
   MEMBER NAME  . . . .  USER01A


 ENTER A NEW PASSWORD TO CHANGE THE TOKEN.

   NEW PASSWORD     ===>
   CONFIRM PASSWORD ===>
```

To create a new token, specify a password in the non-display 'New Password' and 'Confirm Password' fields and hit Enter.

Example: Confirm Delete screen

```
----------------------------  CONFIRM DELETE  ----------------------16:22:55
 ENTER COMMAND ===>                                                     US

 TOKEN TO BE DELETED:
   DATA SET NAME  . . .  'USER02.VERIFY.TOKEN'
   MEMBER NAME  . . . .  USER02A


        SET TOKEN DELETE CONFIRMATION OFF? N  (Y/N)


 PRESS ENTER TO CONFIRM DELETE.
 PRESS END OR RETURN TO CANCEL DELETE.
```

Hit Enter to delete the token. You can set delete confirmation off for the session by specifying 'Y' in the 'Set Token Delete Confirmation Off?' field.

Code Security

Code Security is a password encoding system that allows the CA Verify for CICS REXX component to capture and use passwords while ensuring that they are not disclosed to or used by anyone other than their owner.

Code Security provides effective password protection when:

- A security system is active that reliably establishes the user ID that owns address spaces in which CA Verify for CICS is used, in the manner prescribed by the z/OS System Authorization Facility (SAF).
- User IDs are 2 through 8 contiguous alphanumeric and national characters (@, #, and $).
- Passwords are 3 through 8 contiguous alphanumeric and national characters, can be entered as mixed case characters, and are the first non-null data entered into non-display screen fields.

When a CA Verify for CICS test stream is converted to REXX using Code Security and a password was used in the test stream, the password is replaced by a code in the session data for the REXX script. A code is the encoded value of a password combined with the user ID of the person who owns the password.

The CA Verify for CICS TYPE command CODE keyword identifies a password code in a captured or user-written REXX exec. For example:

```
" TYPE <8,20> CODE '1SJP' ","
<8,24> ENTER "
```

To prevent accidental disclosure of passwords that could occur if codes were incorrectly typed in display fields, the CA Verify for CICS TYPE command for REXX does not allow codes to be typed into display fields.

When a CA Verify for CICS REXX script creates a session and uses a password code to sign on, the code is decoded using the user ID that owns the address space. The correct password can only be recovered from a code in an address space owned by the owner of the code. Because password codes can be seen without disclosing the password encoded and because they can be used only by their owner, codes are not confidential information.

If a password is captured as a code, and system options require password capture notification, the user is notified that a password was captured.

Update Codes When Password Changes

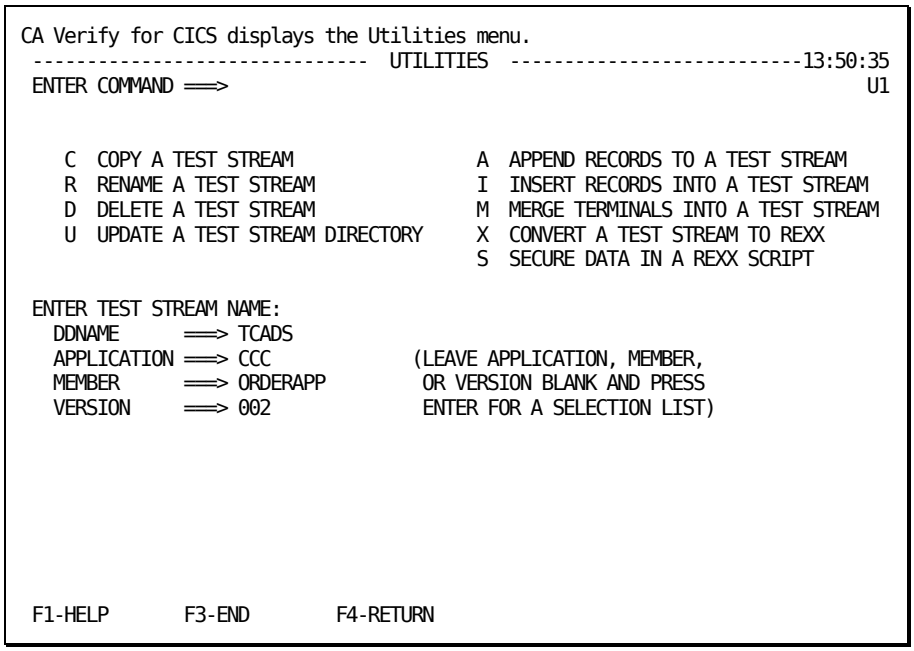
The Code Security Panel is used to create codes to be used in REXX scripts to enter passwords. Because a code contains a password, if a REXX script uses a code for a password and the password is changed, the code must be replaced by a code for the new password, wherever the old code is used.

You can create a new code from the Utilities menu of the TSO interface.

Note: The Code Security Panel is not accessible through the CICS interface.

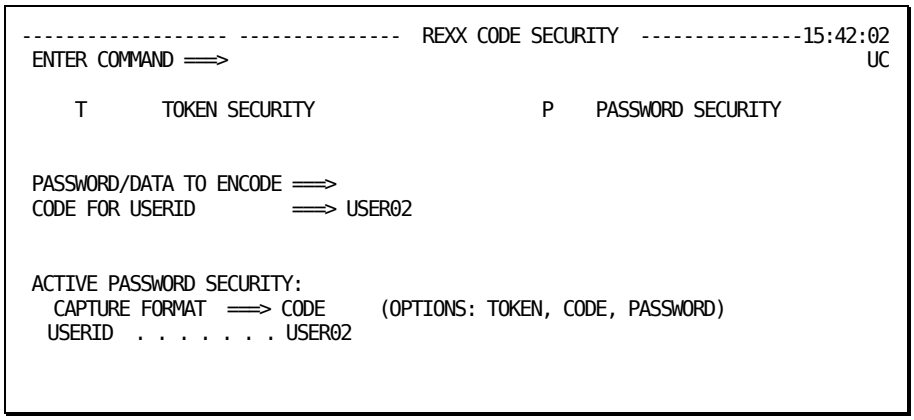
Follow these steps:

1. Type U on the Primary Options Menu of the CA Verify for CICS TSO interface.



2. Select option S. The Code Security panel displays if code security is active for the user. If code security is not active, display the code security panel by typing 'C' on the command line of the displayed security panel and hit Enter.

The code security panel displays:



PASSWORD/DATA TO ENCODE

A non-display field used to enter a password or other confidential data to encode.

CODE FOR USERID

The user ID for which the password is encoded. This field is available only if the CA Verify for CICS REXX options allow codes to be created for use by other user IDs.

CAPTURE FORMAT

The active password capture format used when converting a test stream to a REXX script. If options are listed following the Capture Format field, a listed option may be entered to change the active capture format.

The capture format controls password security for all CA Verify REXX scripts for the user, regardless of when the scripts were created.

USERID

The active user ID. This is a display field only and cannot be modified.

You can perform the following actions on this screen:

- To create a code for password data, enter a password, or other confidential data, in the non-display ‘Password/Data to Encode’ field and hit Enter.
- To create a code for password data for another user ID enter a password, or other confidential data, in the non-display 'Password/Data to Encode' field and enter the user ID in the 'Code for userid' field and hit Enter.
- To change the active capture format for the user ID, specify one of the listed capture format options and hit Enter. The new capture format is stored in the user's profile and the panel for the specified capture format is displayed.

- To display the token or password security panel without changing the active capture format, type '**T**' or '**P**' on the command line and hit Enter.

  Note: The token security and password security options are displayed below the command line if they are available. The options are available if the settings in the REXX options table allow them.

Password Security

Password security neither encrypts nor encodes passwords. It is equivalent to having no security. When a CA Verify test stream is converted to REXX using Password Security and a password was used in the test stream, the actual password is used in the session data for the REXX script and is visible to anyone with the authority to browse or edit the script.

Important! Password security should only be used on test systems which do not require security.

If password security is active, the following panel displays:

```
--------------------------  REXX PASSWORD SECURITY  ------------------15:16:45
 ENTER COMMAND ===>                                                       US

     T   TOKEN SECURITY                    C  CODE SECURITY


   WARNING:

     REXX PASSWORD SECURITY IS NOT ACTIVE WHILE THE FORMAT IS "PASSWORD".
     IF YOU CONVERT A TEST STREAM TO REXX, THE ACTUAL PASSWORD WILL BE
     VISIBLE IN THE REXX SCRIPT.



 ACTIVE PASSWORD SECURITY:
   CAPTURE FORMAT   ===> PASSWORD  (OPTIONS: TOKEN, CODE, PASSWORD)
   USERID  . . . . . . . USER02
```

CAPTURE FORMAT

The active password capture format used when converting a test stream to a REXX script. If options are listed following the Capture Format field, a listed option may be entered to change the active capture format.

The capture format controls password security for all CA Verify REXX scripts for the user, regardless of when the scripts were created.

USERID

The active user ID. This is a display field only and cannot be modified.

You can perform the following actions on this screen.

- To change the active capture format for the user ID, specify one of the listed capture format options and hit Enter.

  The new capture format is stored in the user's profile and the panel for the specified capture format is displayed.

- To display the token or code security panel without changing the active capture format, type '**T**' or '**C**' on the command line and hit Enter.

  Note: The token security and code security options are displayed below the command line if they are available. The options are available if the settings in the REXX options table allow them.

Chapter 11: Batch Functions

This chapter discusses how to use CA Verify for CICS in batch to run, copy, convert, delete, or print test streams and rulesets.

This section contains the following topics:

Overview

Using the batch functions of CA Verify for CICS, you can:

- Copy, delete, and print test streams.
- Define, initialize, and format your data sets easily.
- Perform stress testing at night or on weekends. This lets you avoid periods of peak CICS activity, and keeps testing from interfering with your staff's everyday tasks.
- Perform other important tasks in batch. For example, you can execute CICS in batch and have CA Verify for CICS run your test streams and print the differences.

The sections which follow are divided into MVS and VSE sub-sections where appropriate.

JCL Requirements

This section lists the JCL requirements.

MVS

Use JCL similar to the following example for CA Verify for CICS batch functions.

```
//TCABATCH      EXEC    PGM=TCABATCH,REGION=512K[,PARM='LINECNT=nn']
//STEPLIB       DD      DISP=SHR,DSN=CICS.TCA.LOADLIB
//TCADSIN       DD      DISP=SHR,DSN=CICS.TCADS
//TCADSOUT      DD      DISP=SHR,DSN=CICS.TCADS2
//TCAREXX       DD      DISP=userid.VERIFY.SCRIPT(ssssssss),DISP=SHR
//SYSPRINT      DD      SYSOUT=A
//TCAPRINT      DD      SYSOUT=A
//SYSIN         DD      *
REXX            optional keywords
```

TCABATCH

The EXEC statement specifies program TCABATCH, the load module which performs CA Verify for CICS batch functions. The size of the required region varies depending on the function, but 512K is usually enough. However, CA Verify for CICS may require additional storage when processing multiple terminal test streams.

You can specify a parameter to override the default number of lines per page:

```
PARM='LINECNT=nn'
```

where *nn* is a 2-digit number between 40 and 99. The default is 58.

Note: The batch functions can usually be run in 512 KB. For batch print using display format, allow 512 KB plus 8 KB per terminal.

STEPLIB

Required unless CA Verify for CICS is installed in a linklist library.

TCADSIN

Defines the *input* data set for the Print, Copy, and Directory functions.

TCADSOUT

Defines the *output* data set for the Initialization, Format, Delete, and Copy functions.

TCAREXX

Defines the *output* data set for the Convert to REXX function. The data set may be sequential or partitioned (PDS or PDSE), but must be defined with a fixed record format (F|FB|FBA) and an 80-byte record length. If the data set is partitioned, a member name must be specified.

This DD is only required when using the REXX command to convert a test stream to a REXX exec.

SYSPRINT

Control statements and any error messages are written to this data set. If there is no TCAPRINT DD statement, output from the Print and Directory List functions are also written to SYSPRINT.

SYSPRINT can be allocated to disk, tape, or other sequential device. No DCB information is required. The data set attributes are fixed blocked 133-byte records with ASA carriage control. CA Verify for CICS selects the largest blocksize applicable to the device.

TCAPRINT

If specified, output from the Print and Directory Listing functions is written to this data set. If omitted, this output is written to SYSPRINT.

SYSIN

Defines the *control statement input* for batch functions. It can be allocated to disk, tape, or other sequential device. The record length can exceed 80 bytes.

VSE

Use JCL similar to the following example for CA Verify for CICS batch functions.

```
// JOB          RUN CA VERIFY BATCH FUNCTIONS
// DLBL         VERLIB,'CICS.VERIFY'
// EXTENT       ,CICS01
// LIBDEF       *,SEARCH=VERLIB.VERIFY
// DLBL         IJSYSUC,'VSAM.USERCAT',,VSAM
// DLBL         TCADSIN,'CICS.TCADS1',,VSAM
// DLBL         TCADSOU,'CICS.TCADS2',,VSAM
// ASSGNSYSIPT,00C
// ASSGNSYSLST,00E
// EXEC         PGM=TCABATCH,SIZE=512K[,PARM='LINECNT=nn']
DIRECTORY
/*
/&
```

TCABATCH

TCABATCH is the load module which performs batch functions. The size of the required region varies depending on the function, but 512 KB is usually enough. However, CA Verify for CICS may require additional storage when processing multiple terminal test streams.

You can specify a parameter to override the default number of lines per page:

```
PARM='LINECNT=nn'
```

where *nn* is a 2-digit number between 40 and 99. The default is 58.

Note: The batch functions can usually be run in 512 KB. For batch print using display format, allow 512 KB plus 8 KB per terminal.

LIBDEF

Required unless CA Verify for CICS is installed in a permanently assigned core image library.

TCADSIN

This DLBL statement defines the *input* data set for the Print, Copy, and Directory functions.

TCADSOU

This DLBL statement defines the *output* file for the Initialization, Format, Delete, and Copy functions.

SYSLST

Control statements, any error messages, and output from the Print and Directory List functions are written to this file.

SYSLST cannot be allocated to disk; tape or other sequential devices are acceptable. The file attributes are fixed unblocked 133-byte records with ASA carriage control.

SYSIPT

Defines the *control statement inp*ut for batch functions. It can be allocated to disk, tape, or other sequential device. The record length can exceed 80 bytes.

Control Statement Format

Observe the following rules when coding control statements:

- Use columns 1-72.
- Blank statements and statements with an asterisk in column 1 are printed but not processed.
- Many control statements and parameters can be abbreviated. In the following sections, the minimum abbreviation is underlined.

- All batch functions can be specified in the same execution.
- It is not necessary to continue control statements. However, the Print command can effectively be continued by specifying two or more consecutive Print commands.

Define and Initialize Data Sets

This section defines and initializes data sets.

Define Data Sets (MVS)

Use JCL similar to the following example to define a data set.

```
//DEFINEIT      EXEC    PGM=IDCAMS
//SYSPRINT      DD      SYSOUT=A
//SYSIN         DD      *
        DEFINE  CLUSTER ( -
                        NONINDEXED                      /* REQUIRED */ -
                        SHAREOPTIONS(2 3)               /* OR (3 3) */ -
                        SPEED -
                        UNIQUE -
                        CYLINDERS(5 5) -
                        VOLUMES(VSAM01) -
                        NAME('CICS.TCADS') -
                        RECORDSIZE(16377 16377)  /* MUST BE CISIZE-7 */ -
                        CONTROLINTERVALSIZE( 16384) )
```

- The first parameter is required.
- The space allocation must be large enough for at least 11 control intervals.
- The control interval size must be a multiple of 4096 up to a maximum size of 28,672. The recommended size is 16,384.
- The record size must be 7 less than the control interval size.
- The other parameters have no special requirements.

Note: Use SHAREOPTIONS (2 3) if the data set is only being used in one region. Use SHAREOPTIONS (3 3) if the data set will be updated from two regions.

Define Files (VSE)

Use JCL similar to the following example to define a CA Verify for CICS file.

```
// DLBL IJSYSUC,'VSAM.USERCAT',,VSAM
// EXEC         PGM=IDCAMS,SIZE=AUTO
        DEFINE  CLUSTER ( -
                NONINDEXED                      /* REQUIRED */ -
                SHAREOPTIONS(2)                 /* REQUIRED */ -
                SPEED -
                UNIQUE -
                CYLINDERS(5 5) -
                VOLUMES(VSAM01) -
                NAME('CICS.TCADS') -
                RECORDSIZE(16377 16377)  /* MUST BE CISIZE-7 */ -
                CONTROLINTERVALSIZE(16384) ) -
                CATALOG(VSAM.USERCAT)
/*
```

- The first two parameters are required.
- The space allocation must be large enough for at least 11 control intervals.
- The control interval size must be a multiple of 4096 up to a maximum size of 28,672. The recommended size is 16,384.
- The record size must be 7 less than the control interval size.
- The other parameters have no special requirements.

Initialize Data Sets (MVS and VSE)

You must use the TCABATCH Init function after you have used IDCAMS to define the data set or file. This function initializes the data set allocated to TCADSOUT (for MVS) or TCADSOU (for VSE).

The format is:

```
INIT
```

This function has no parameters.

See the JCL Requirements section earlier in this chapter for the necessary JCL.

Format Data Sets (MVS and VSE)

Use the TCABATCH Format function to format control intervals on the data set allocated to TCADSOUT (MVS) or TCADSOU (VSE). Although CA Verify for CICS will automatically format control intervals as needed, it is more efficient to format control intervals in batch when CA Verify for CICS is not in use.

Specify the Format card in the SYSIN DD data set. The format is:

```
FORMAT
```

Directory Listing of Test Streams and Rulesets

The Directory function lists the test streams and rulesets in the data set allocated to TCADSIN (MVS and VSE).

Specify the Directory card in the SYSIN DD data set. The format is:

```
DIRECTORY
```

This function has no parameters.

The Directory listing provides the same test stream information as the Directory Information and Initial Terminal Status menus. See the "Browse Function" chapter for a discussion of the fields.

The same ruleset information that appears on the Browse: Ruleset Directory panel is also provided.

List Totals and Averages for All Test Streams

After listing information for each test stream, the Directory listing includes the following statistics for the entire data set:

- Total number of test streams
- Total number of input and output screens
- Average input and output screen size

- VSAM control interval size
- Formatted control intervals; that is, the number formatted by the Format function or formatted as needed by CA Verify for CICS
- Number of control intervals currently in use
- Unused control intervals
- Unavailable control intervals

Note: The *unavailable* control intervals cannot be used because of system failures during data set updates. Usually this number will be zero; it should always be small. If this number is large, you can recover the space by using the file reorganization procedure. The file reorganization procedure is described in the Copy Test Streams section.

Select Test Streams and Rulesets to Copy, Convert, Print, and Delete

Use the Select function to specify the test streams and rulesets to be copied, printed, or deleted. One or more Select cards must follow a Copy, Print, or Delete card. The format is:

```
SELECT name name name ... name
```

- Specify one or more names separated by blanks. When you specify a name, *both* the test stream *and* the ruleset having that name are selected. When using the REXX function to convert a test stream to a REXX exec, only one test stream may be specified.
- Specify each name as follows:

  ```
  application.member.version
  ```

  For example:

  ```
  SELECT GENERAL.TEST.001 GENERAL.TEST.002
  ```

  Selects the test streams and rulesets named GENERAL.TEST.001 and GENERAL.TEST.002.

Specify Names Generically

To specify names generically, use an equal sign to represent an actual character. For example:

```
SELECT GENERAL.TEST.===
```

Selects all versions of the test stream with an Application of GENERAL and a Member of TEST.

```
SELECT GENERAL.========.001
```

Selects version 001 of all test streams and rulesets with an Application of GENERAL, regardless of the Member name.

```
SELECT GENERAL.=.===
```

Selects all test streams and rulesets with an Application of GENERAL, providing the Member name is only one character. CA Verify for CICS pads with blanks to the end of each section of a generic name.

```
SELECT =X======.========.===
```

Selects all test streams and rulesets that have X as the second character of the Application.

You can also specify that CA Verify for CICS select the *highest* version of a test stream by substituting >>> for the version number. For example:

```
SELECT GENERAL.TEST.>>>
```

Selects the highest version of the GENERAL.TEST test stream.

Select Terminals for Copy and Print Functions

Use the Terminals function to specify terminals to be copied or printed. The Terminals card applies only to test streams. One or more Terminals cards may precede a Copy or Print card. The format is:

```
TERMINALS name name name ... name
```

Specify one or more terminal names, separated by blanks. For example:

```
TERMINALS TRM1 TRM2
```

selects terminals TRM1 and TRM2.

You can specify consecutive Terminals cards to create a cumulative list of terminals to be copied or printed. Non-consecutive Terminals cards create separate lists. For example:

```
TERMINALS TRM1 TRM2
PRINT
SELECT GENERAL.TEST.001
TERMINALS TRM3 TRM4
COPY
SELECT GENERAL.TEST.002
```

In this case, CA Verify for CICS will print screens for terminals TRM1 and TRM2 for test stream GENERAL.TEST.001, and copy screens for terminals TRM3 and TRM4 for test stream GENERAL.TEST.002.

Copy Test Streams and Rulesets

Use the Copy function to copy test streams and rulesets from the data set allocated to TCADSIN to the data set allocated to TCADSOUT (MVS) or TCADSOU (VSE).

Specify the Copy card in the SYSIN DD data set. The format is:

```
COPY
```

This function has no parameters. One or more Select cards should follow the Copy card. See Select Test Streams for Copy, Convert, Print, and Delete (see page 254) earlier in this chapter for details.

You can also specify one or more Terminals cards to indicate which terminals are to be copied. See Select Terminals for Copy and Print Functions earlier in this chapter for details.

Reorganize the Data Set

The copy function can be used to migrate an existing data set to a larger data set or to reclaim unavailable control intervals. Usually reorganization is not required.

A data set cannot be reorganized in place.

Follow these steps:

1. Define a new data set with IDCAMS.

2. Run TCABATCH with TCADSIN allocated to the old data set and TCADSOUT (MVS) or TCADSOU (VSE) allocated to the new one. Use the following control cards to initialize and format the new data set and copy the test streams:

   ```
   INIT
   FORMAT
   COPY
   SELECT ========.========.===
   ```

3. Delete the old data set and rename the new one.

Convert a Test Stream to REXX

Use the REXX function to convert a test stream from the data set allocated to TCADSIN to the REXX script file allocated to TCAREXX. The resulting REXX exec can be modified as necessary to meet user needs.

Note: The password security format used in the converted REXX script (that is token, code, or password) is determined by the password security option that is active for the userID that performs the conversion, at the time the test stream is converted to REXX. See the section Secure Data in a REXX Script (see page 233) for more information on the security options, including viewing or modifying the active security option for your userID.

Specify the REXX card in the SYSIN DD data set. The format is:

```
REXX APPLID(applid) SLINES(nn) [DELAY/NODELAY] [PROTECT/NOPROTECT]
```

APPLID(applid) (Optional)

Specifies the VTAM applid of the application to which the converted REXX exec should establish a session when it is executed. If APPLID is omitted, a "LOGON '????????'" statement will be generated by the conversion routine and the user must modify the exec to provide a valid applid before executing it.

SLINES(nn) (Optional)

Specifies the number of lines of an output screen to be included in the converted REXX exec. The default value is 3. The screen lines are included as comments to aid readability of the exec.

DELAY/NODELAY

Specifies whether DELAY commands should be added to the converted REXX exec, to reflect the think time associated with the original test stream. The default is DELAY.

PROTECT/NOPROTECT

Specifies whether the REXX script (indicated by the TCAREXX DD) should be overwritten if it already exists. If PROTECT is specified and the target script exists, the member will not be overwritten. Instead, a message will be written to SYSPRINT. If NOPROTECT is specified and the target member exists, it will be overwritten. The default is NOPROTECT.

A Select card should follow the REXX card. The Select card specifies the test stream to be converted. See the section, Select Test Streams for Copy, Convert, Print, and Delete (see page 254) in this chapter for details.

Sample JCL

Use JCL similar to the following example to convert a test stream to REXX in batch..

```
//TCABATCH      EXEC    PGM=TCABATCH,REGION=2048K
//STEPLIB       DD      DISP=SHR,DSN=yourHLQ.CATJLOAD
//TCADSIN       DD      DISP=SHR,DSN=hlq.TCADS
//TCADREXX      DD      DISP=SHR,DSN=userid.VERIFY.SCRIPT(ssssssss)
//SYSPRINT      DD      SYSOUT=A
//TCAPRINT      DD      SYSOUT=A
//SYSIN DD      *
REXX APPLID(vtamappl) SLINES(24)
SELECT application.member.version
/*
```

Delete Test Streams and Their Rulesets

Use the Delete function to delete test streams and rulesets from the data set allocated to TCADSOUT (MVS) or TCADSOU (VSE).

Specify the Delete card in the SYSIN DD data set. The format is:

```
DELETE
```

This function has no parameters. One or more Select cards should follow the Delete card. The Select card will select both the test streams and the rule sets that match it. For details, see the section Select Test Streams and Rulesets to Copy, Print, and Delete that appeared earlier in this chapter.

Print Test Streams and Rulesets

Use the Print function to print test streams and rulesets from the data set allocated to TCADSIN.

Specify the Print card in the SYSIN DD data set. The format is:

```
PRINT
```

One or more Select cards should follow the Print card. See the section Select Test Streams and Rulesets to Copy, Print, and Delete earlier in this chapter for details. You can also specify one or more Terminals cards to indicate which terminals are to be printed. See Select Terminals for Copy and Print Functions earlier in this chapter for details.

For each *test stream* selected, the Print function lists directory information, terminal status information, and the screens. See the "Browse Function" chapter for an explanation of directory and terminal status information.

For each *ruleset* selected, a summary like the Rules—Summary panel shown on page 8-8 will be printed. See the "Rules Function" chapter for an explanation of the rules information.

Parameters for the Print function follow. All of the following parameters see how the *test stream* will be printed. Mutually exclusive parameters are separated by slashes, with the default listed first. If mutually exclusive parameters are specified, CA Verify for CICS uses the parameter specified last. Overrides to defaults remain in effect until CA Verify for CICS encounters the next Print card. However, consecutive Print cards are considered a continuation and overrides remain in effect.

Format Parameters

DISPLAY/DUMP/FIELD

Specifies screen format.

Display

The screen appears as it appeared at the terminal during logging — for 3270 non-graphic test streams.

Dump

The screen appears in dump format — for test streams created at any terminal. Minimal formatting is provided. The header information includes the record number, data length, terminal name, for example, followed by the data stream sent to or received from the terminal. If specified or left as the default, the hexadecimal translation of the data stream appears. For 3270 test streams, the data stream includes the Write Control Character (WCC) for output screens and the attention identifier (AID) and cursor location for input screens.

Field

Each field in the data stream is listed on a separate line — designed for 3270 graphic or non-graphic test streams, but supports any terminal.

See the "Run Function" chapter for a detailed description of the different formats.

Note: You can specify Display format even if a test stream has mixed terminal types. When necessary, the Print function will print screens in Field or Dump format.

General Parameters

The following parameters apply to all three formats.

RULER/NORULER

Determines whether or not a column number ruler should be printed for the terminal control table user area and data in Field and Dump formats.

Ruler

A ruler is printed

Noruler

A ruler is not printed

NOHEX/HEX

Specifies whether or not the hexadecimal equivalent of each character should be printed on the next two lines for the terminal control table user area and data in Field and Dump formats.

Nohex

The hex equivalent is not printed. This is the default for Display and Field formats.

Hex

The hex equivalent is printed. This is the default for Dump format.

NOINPUT/NOOUTPUT

Controls whether input or output screens are printed.

Noinput

Only output screens are printed.

Nooutput

Only input screens are printed.

If this parameter is omitted, both input and output screens are printed.

Display Parameters

The following parameters apply *only* to Display format.

FRAME/NOFRAME

Specifies whether or not a frame of dashes surrounds the screen.

Frame

A frame is printed.

Noframe

A frame is not printed.

CENTER/NOCENTER

Specifies whether or not the screen is centered on the page.

Center

The screen is centered.

Nocenter

The screen is left justified. Use this option when you intend to photocopy the page.

HIGH/NOHIGH

Specifies whether or not high-intensity fields are overprinted.

High

High-intensity fields are overprinted. Do not specify this parameter if the output is sent to a laser printer or a printer which does not support overprinting.

Nohigh

High-intensity fields are not overprinted.

NOULINE/ULINE

Specifies whether or not unprotected fields are underlined.

Nouline

Unprotected fields are not underlined.

Uline

Unprotected fields are underlined by overprinting.

NOLOW/LOW

Specifies whether or not low-intensity (non-display) fields are printed.

Nolow

Low-intensity fields are not printed.

Low

Low-intensity fields are printed.

Note: For test streams logged with the DRKPROT=YES installation option in effect, input data entered in low-intensity fields will *not* be printed even if LOW is specified.

NOSHRINK/SHRINK

Specifies whether or not blank screen rows should be skipped in order to reduce the size of the listing.

Noshrink

Blank screen rows are printed.

Shrink

Blank screen rows are skipped. The printed rows are numbered.

NOSTATS/STATS

Specifies whether or not statistics for the current screens should be printed.

Nostats

Statistics are not printed.

Stats

Statistical information, including record number, type of operation, data stream length, screen size, and so on, is printed.

CURSOR(_ _)/CURSOR/NOCURSOR

Specifies how the cursor position should be handled.

Cursor (_ _)

One or two characters — e.g., underscores, the default — will be overprinted to indicate cursor position. Do not specify this parameter if the output is sent to a laser printer or a printer which does not support overprinting.

Cursor

The cursor row and column will be printed under each screen.

Nocursor

The cursor position will be not be indicated.

Printing Parameters

Depending on the options specified when a test stream is run, the output test stream may contain mismatch and applied rules information. The following parameters control the printing of this information.

APRULES(YES)/(NO)/(ONLY)

Controls the printing of the rules applied before the mismatch occurred.

APRules(Yes)

Print the Rules Applied Before Mismatch screens while printing the test stream.

APRules(No)

Do not print the Rules Applied Before Mismatch screens while printing the test stream.

APRules(Only)

Print only those screens that had a Rules Applied Before Mismatch screen created by the Run.

DIFF(NO)/(YES)/(ONLY)

Controls the printing of mismatch information recorded when the Run Record History option is set at Y.

DIFF(No)

Differences are not printed.

DIFF(Yes)

Differences are printed.

DIFF(Only)

Only those screens that had a difference are printed.

SEPARATE(EXPECTED/CURRENT/NONE)/COMBINED

Controls the format of mismatch information.

Separate

Prints information in this order: Expected screen, current screen, Differences, and Applied Rules. Separate is the default.

Separate(Expected)

Prints everything except the Current screen.

Separate(Current)

Prints everything except the Expected screen.

Separate(None)

Only the differences and the Applied Rules are printed.

Combined

One screen is printed. This screen contains one row for each row that is the same and three rows for each row that is different. The rows are labeled:

E/C

Expected and current — for rows that are the same

Expt

Expected — the row from the expected screen when the rows are different

Curr

Current — the row from the current screen when the rows are different

Diff

The differences between the rows

Blank

The characters are equal

X

The characters are not equal

-

The characters are within a variable field

Regardless of your specifications, Combined format uses the following Print function parameters: Nohigh, Low, Nouline, Nocenter, Noshrink, and Cursor.

SIGNOFF(NO)/(YES)/(ONLY)

Controls printing of mismatch signoff information recorded when the Run Require Signoff Data option is set to Y.

Signoff(No)

Mismatch signoff information is not printed.

Signoff(Yes)

Mismatch signoff information is printed.

Signoff(Only)

Only screens with a mismatch are printed.

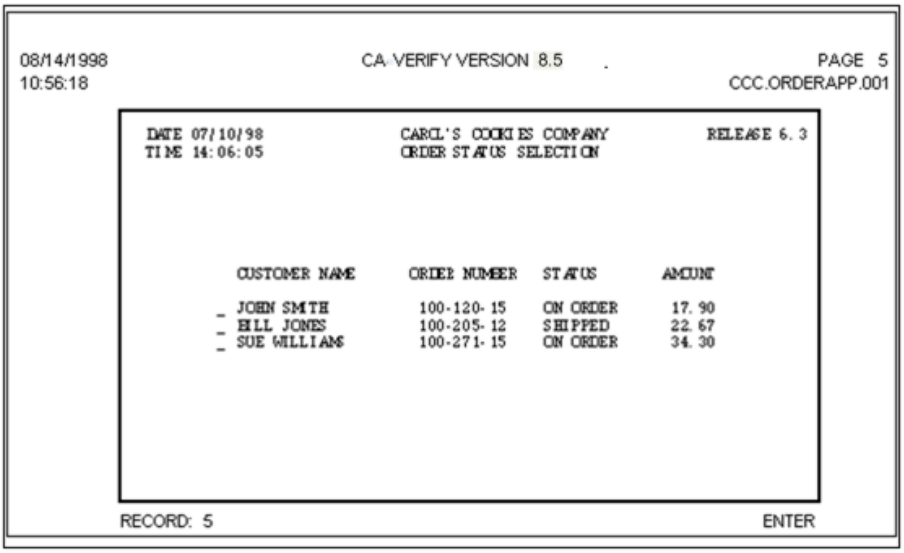
If signoff information was specified, it is printed to the right of the screen rows which had a mismatch, or on a separate row if more room is needed.

If signoff information was not specified, rows of underscores are printed so the signoff information can be inserted.

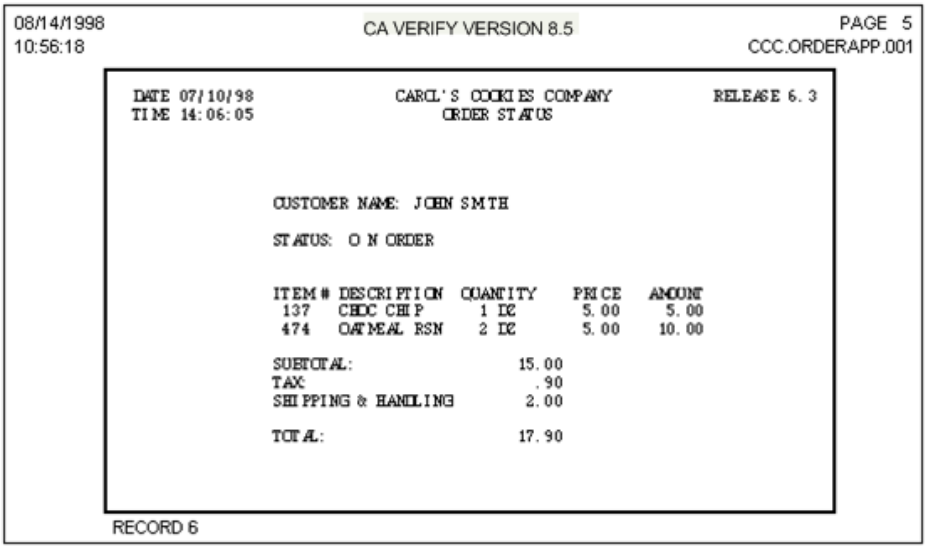
Examples

The examples on the next pages illustrate output from the Print function. The same input and output screens are illustrated in the three formats: Display, Field, and Dump.

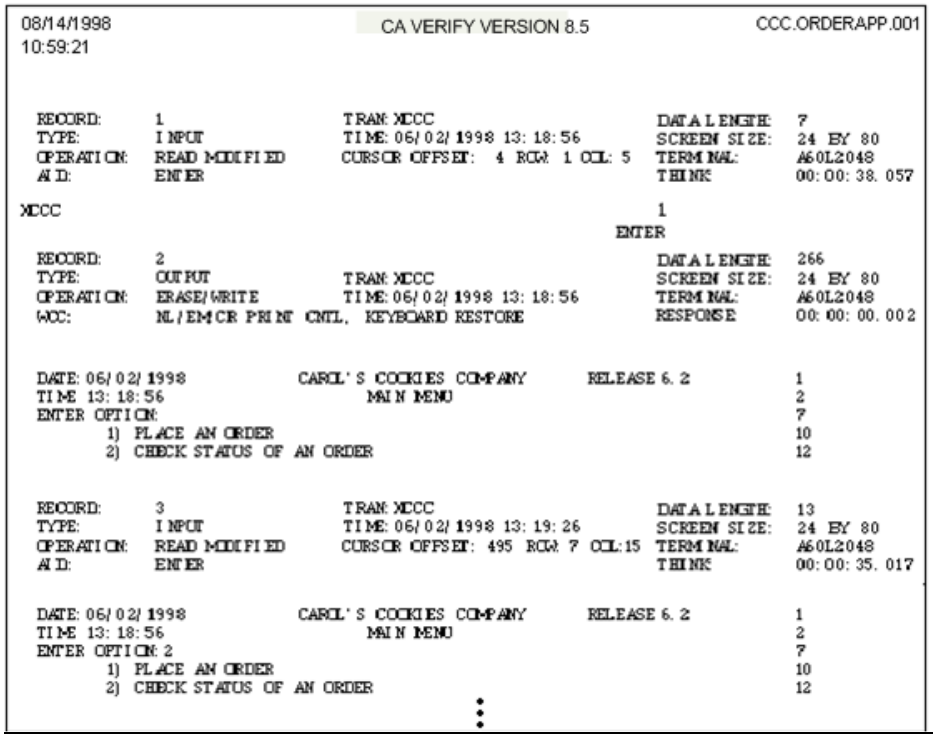
This input screen was printed in Display format with the default options:



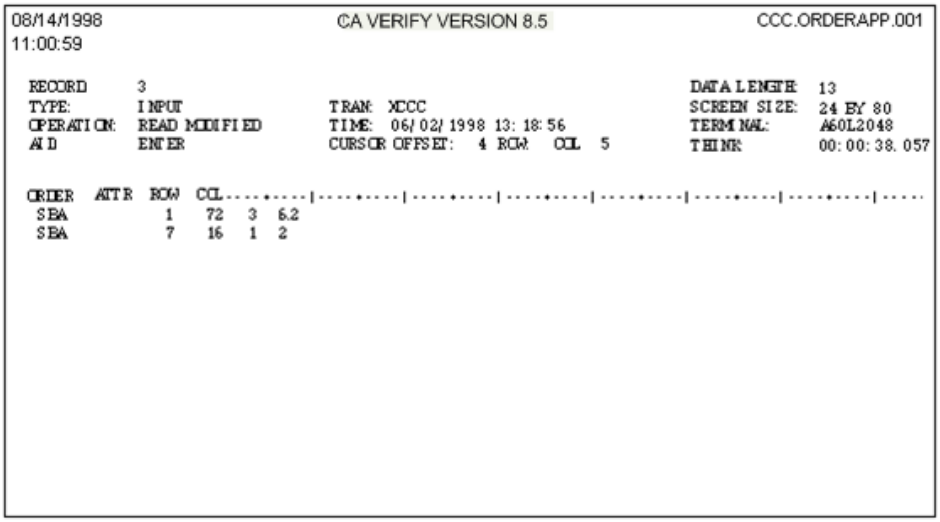
This output screen was printed in Display format with the default options:



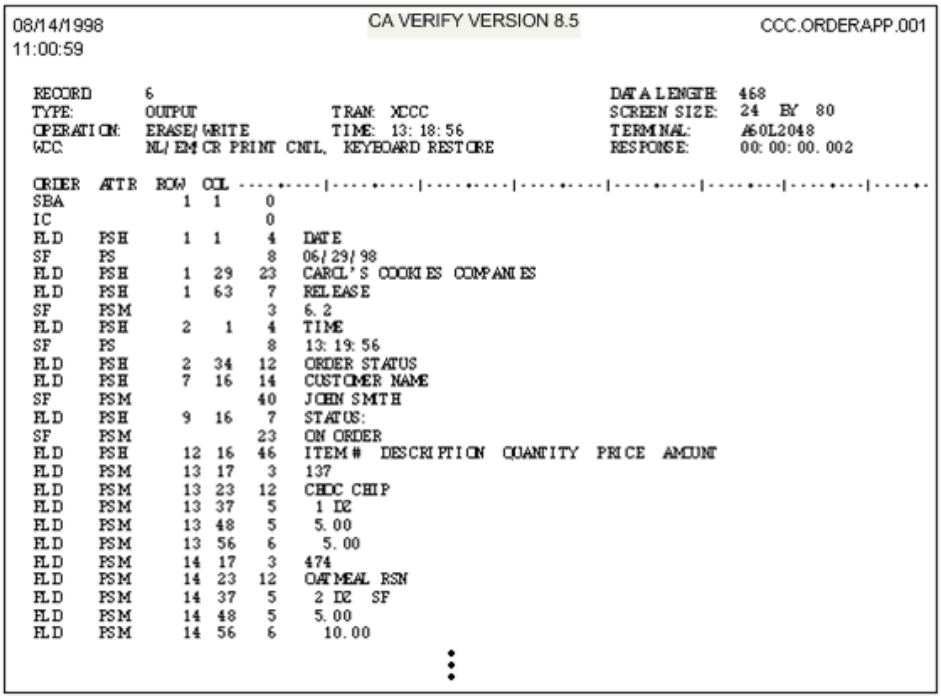
This screen was printed in Display format with the Shrink, Nocenter, Stats, Noframe, and Nocursor options:



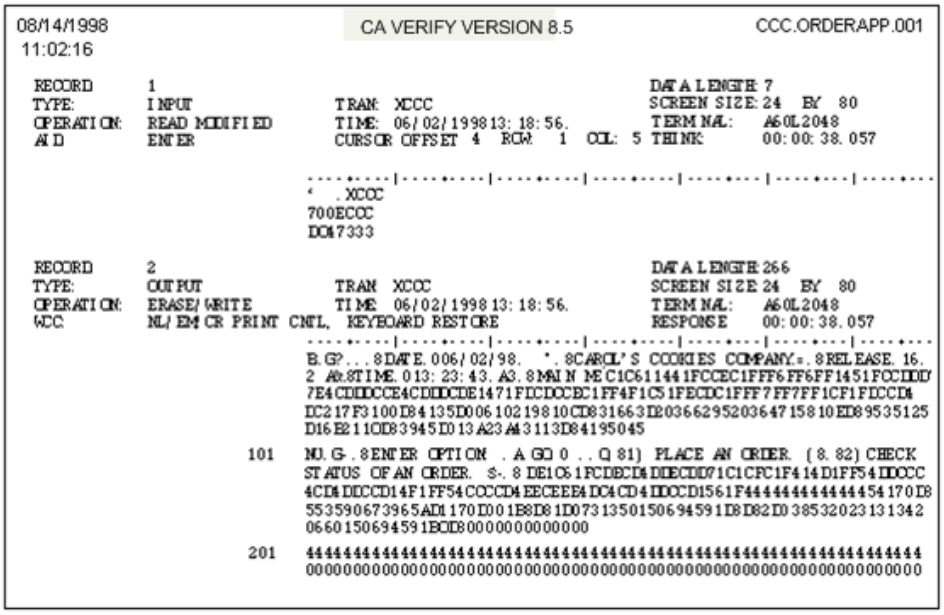
This input screen was printed in Field format with the default options:



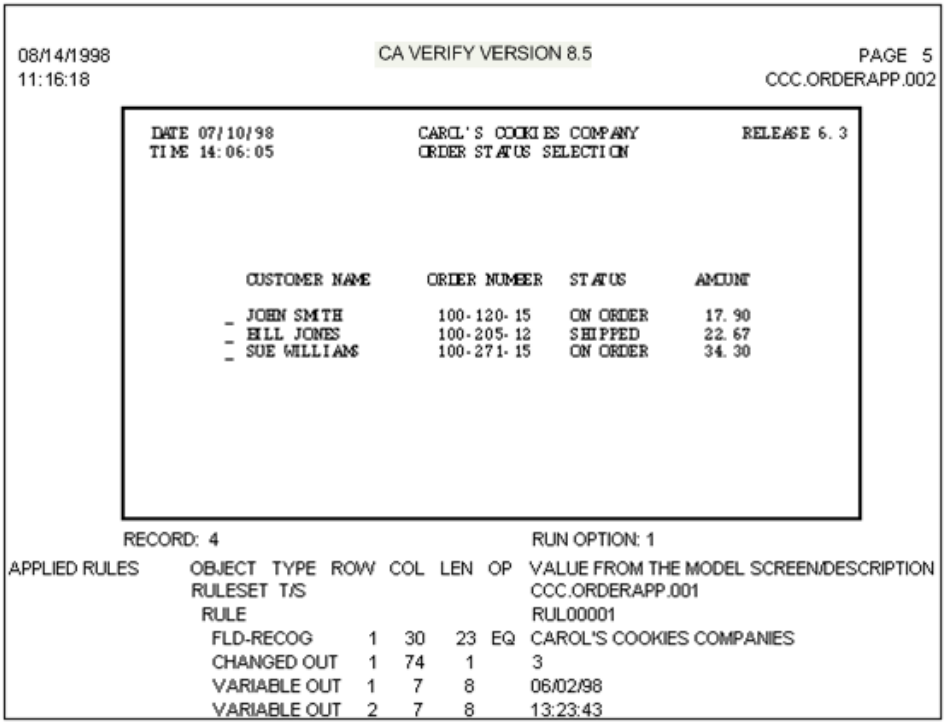
This output screen was printed in Field format with the default options:



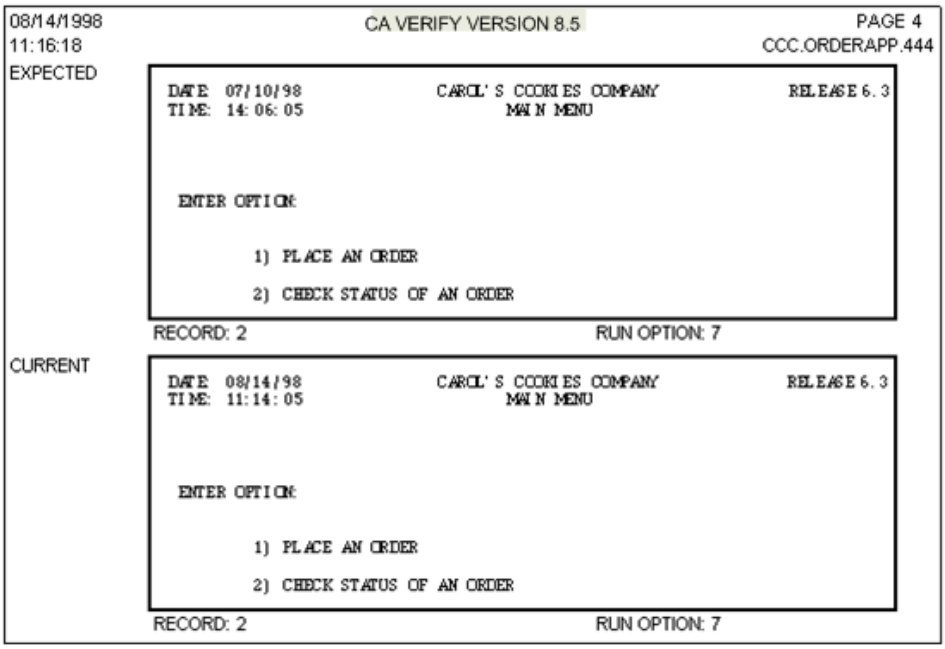
The following screen was printed in Dump format with the hex option.



This screen was printed with Applied Rules:



The next screens printed contain Differences:



Batch Run

The Batch Run function executes CICS and runs test streams in batch. Differences between the expected and current screens can then be printed using the Print function.

To run CICS in batch, the TCT should contain:

- A sequential terminal (disk or CRLP)
- CA Verify for CICS virtual terminals

Real terminals may be included but are not required.

CICS batch execution can be controlled entirely from the sequential terminal. The transactions for the sequential terminal should include:

1. A signon transaction
2. CA Verify for CICS transactions to run the test streams
3. A shutdown transaction

The format for the CA Verify for CICS transactions to invoke the Run function from a sequential terminal is:

XTCA RUN d.a.m.v. THINK(NONE) LIMIT(999999) [SCREEN/LOGICAL/PHYSICAL/NONE] SUPOTS [BATSIGN = xxxxxxxx]

XTCA Run (Required)

Invokes the run.

d.a.m.v. (Required)

Identifies the test stream:

d

DDname

a

Application

m

Member

v

Version

Think(NONE) (Optional)

Specifies no simulated operator think time, the default. You can also specify a number of seconds or a percentage. See the "Run Function" chapter for valid parameters.

Limit(nnnnnn) (Optional)

Specifies the maximum number of mismatches CA Verify for CICS should allow. The default is 999999. If this number is exceeded, CA Verify for CICSy stops the run. Mismatches generated up to that point are recorded and the output test stream is retained.

Screen/Logical/Physical (Optional)

Specifies the type of comparison. The default is screen. See the "Run Function" chapter for an explanation of these options.

Supots (Optional)

The default is for CA Verify for CICS to create an output test stream with the same name as the input test stream and the version incremented to the next available one. Mismatch data is automatically recorded. Specify this option to suppress the output test stream.

[Batsign = xxxxxxxx] (Optional)

Specifies an override for the XTCAOPTS installation option AUTSIGN = APPLICATION. xxxxxxxx is the name of the #SIGNON and #SIGNOFF test stream. The BATSIGN parameter value will be substituted for the application when the #SIGNON and #SIGNOFF test streams are run.

Processing during a Batch Run

During a batch run, CA Verify for CICS:

- Accepts missing outputs
- Does not perform mismatch confirmation
- Accepts I/O mismatches; that is, replaces the original screen with the current screen in the output test stream

MVS JCL

Use JCL similar to the following example to run test streams in batch.

```
//VERIFYJOB ...
//CICS          EXEC    DFHSIP
//TCADS         DD      DISP=SHR,DSN=CICS.TCADS
//CR            DD      *       Card Reader Half of CRLP – Notes
CSSN NAME=OVERNIGHT,PS=NITE \
XTCA RUN TCADS.GENERAL.PAYROLL.001 THINK(50%) \
XTCA RUN TCADS.GENERAL.CLAIMS.003 LIMIT(100) PHYSICAL \
CSMT SHUT,Y \
/*
//LP            DD      SYSOUT=ACA Verify Completion Messages
.
.
. normal CICS startup JCL
.
.
.
//REPORTEXEC    PGM=TCABATCH
//STEPLIB       DD      DISP=SHR,DSN=CICS.TCA.LOADLIB
//TCADSIN       DD      DISP=SHR,DSN=CICS.TCADS
//SYSPRINT      DD      SYSOUT=A
//TCAPRINT      DD      SYSOUT=A
//SYSIN         DD      *
PRINT DIFF (ONLY)
SELECT GENERAL.PAYROLL.>>>
SELECT GENERAL.CLAIMS.>>>
/*
```

In this example:

- The CICS step performs the batch run; the Report step prints the output.
- The Report step can be executed without a CICS step to print the results of runs for which the Run Record History option was "Y". See the Print Function earlier in this chapter.
- The Select cards in the Report step specify versions of >>> which instructs CA Verify for CICS to select the highest versions of the test streams.

Note:

- The CR DD statement must point to a data set whose blocksize is 80.
- A delimiter must appear after the last character of each CR DD input statement. This delimiter must be the one specified in the EODI operand in the TCP generation.

VSE JCL

Use JCL similar to the following example to run test streams in batch.

```
// JOB          CICS
// DLBL         VERLIB,'CICS.VERIFY
// EXTENT       ,CICS01
// LIBDEF       *,SEARCH=VERLIB.VERIFY
// DLBL         IJSYSUC,'VSAM.USERCAT',,VSAM
// DLBL         TCADS,'CICS.TCADS',,VSAM
// ASSGNSYSIPT,00C      (CR)
// ASSGN        SYSIPT,00E      (LP)
.
.
. normal CICS startup JCL
.
.
// EXEC         DFHSIP
CSSN NAME=OVERNIGHT,PS=NITE \
XTCA RUN TCADS.GENERAL.PAYROLL.001 THINK(50%) \
XTCA RUN TCADS.GENERAL.CLAIMS.003 LIMIT(100) PHYSICAL \
CSMT SHUT,Y \


*STEP2: PRINT RESULTS
// DLBL         VERLIB,'CICS.VERIFY'
// EXTENT       ,CICS01
// LIBDEF       *,SEARCH=VERLIB,VERIFY
// DLBL         IJSYSUC,'VSAM.USERCAT',,VSAM
// DLBL         TCADSIN,'CICS.TCADS',,VSAM
// ASSGNSYSIPT,00C
// EXEC         PGM=TCABATCH,SIZE=512K
PRINT DIFF (ONLY)
SELECT GENERAL.PAYROLL.>>>
SELECT GENERAL.CLAIMS.>>>
/*
```

- The first step performs the batch run; the second step prints the output.
- The Print function can be executed without the CICS step to print the results of runs for which the Run Record History option was "Y". See the Print Function earlier in this chapter.
- The Select cards specify versions of >>>, which instructs CA Verify to select the highest version of the test stream.

Note: A delimiter must appear after the last character of each SYSIPT statement. This delimiter must be the one specified in the EODI operand in the TCP generation.

Chapter 12: Commands for REXX

This chapter describes the CA Verify for CICS Host Command Environment which allows REXX execs to ADDRESS VERIFY and use virtual terminals in a multi-session synchronous or asynchronous environment.

The CA Verify for CICS Host Command Environment allows REXX execs to ADDRESS VERIFY and to issue CA Verify for CICS host commands.

- CA Verify for CICS LOGON, TYPE, INVITE, DELAY, and LOGOFF commands allow REXX execs to start sessions using virtual terminals and to use online software that was designed for interactive use by people using 3270 terminals. Using CA Verify for CICS, REXX execs that repeat captured sessions may be created automatically by converting previously logged test streams to REXX execs.
- CA Verify for CICS TRACE and ENDTRACE commands allow execs to capture exec-driven sessions in datasets. Using CA Verify for CICS Interactive System functions, traced virtual terminal screen images may be displayed, printed, or used to perform regression testing.
- The ATTACH command creates new z/OS tasks and REXX Language Processor Environments and uses the environments to execute REXX execs that run at the same time as the attaching exec. Attached exec tasks can start and drive asynchronous sessions to conduct stress tests or to multiply throughput in production applications.
- The ACCESS command allows an exec to use multiple sessions, to attach multiple exec tasks, and to use multiples of any of the application's resource. Automatic resource locks, LOCK and UNLOCK commands allow multiple execs to access and share any of the application's resource.
- QUEUE, PUSH and POST commands allow execs to store data in designated queues to communicate with any number of other asynchronous exec tasks that process queued data using PULL or QUERY QUEUE commands.

- QUERY, HANDLE, WAIT and SIGNAL commands allow execs to inquire about the applications' resources, to modify or interrupt command processing, and to process errors and other conditions using REXX condition traps.
- The MONITOR command activates integrated monitor functions that display the application's commands, command functions, exec-driven sessions, and error messages automatically, and may be used to debug execs or to display multiple virtual terminal screens at a single terminal to monitor concurrent session activity driven by asynchronous attached execs while performing a system stress test.

This section contains the following topics:

The Session Commands

CA Verify for CICS session commands may be used by REXX execs to start sessions with online systems using virtual terminals and to use virtual terminals in the same way that people use real terminals to access online system software.

- The LOGON command allocates a virtual terminal and uses the terminal to start a session with any online system that supports terminal sessions.
- The TYPE command uses the virtual terminal keyboard to key in data, position the cursor, and to send data by pressing the Enter key, PF keys or other keys that send data.
- The INVITE command waits for an online system to send anticipated messages, like sign on prompts at the start of a session.

- The DELAY command simulates human think time and controls the speed of an exec driven session.
- The LOGOFF ends a session and frees a virtual terminal.

CA Verify for CICS session commands return information to execs in REXX variables. At the completion of each session command, a copy of the virtual terminal display buffer is returned in a REXX variable, PTEBUFF, the virtual terminal screen size and cursor position are stored in PTEROWS, PTECOLS and PTECSRP, and other information about the virtual terminal and session is returned in other REXX variables. Using the returned display buffer, execs may analyze transaction responses and gather displayed information or use it to enter subsequent transactions.

REXX execs that repeat captured sessions may be created automatically by converting previously logged test streams to execs. When a logged test stream is converted to REXX, session initiation is captured as a LOGON command and user keystrokes are captured as TYPE commands. Converted execs also include INVITE commands that wait for anticipated system messages and may include DELAY commands that capture actual human think-time delays that occur while logging a test stream. A converted exec may be executed to repeat a captured session automatically.

By modifying a captured exec or by writing execs from scratch, intelligent REXX execs may be developed that use the application's sessions and any available online system software to reliably perform or automate almost any imaginable terminal task that could by performed by a person using a terminal.

The REXX exec in the following example uses the application session commands to start a session and to use transactions designed for users.

```
     /* Exec: OPEN CA Verify commands for REXX: */
     /* LOGON INVITE TYPE LOGOFF */
(1)      address Verify
(2)    "  logon cics 24x80 24x80 basicds "
       "  invite 1000 "
       "  type clear "
(3)    "  type 'cemt set dataset(somedd) open' enter "
(4)      if substr(PTEBUFF,224,6) = 'NORMAL' then
            say 'open worked ok'
         else
            say 'open failed!!!'
       "  type pf3 "
(5)    "  logoff "
```

Tracing Exec-Driven Sessions

CA Verify for CICS TRACE and ENDTRACE commands can be used to capture an exec-driven CA Verify for CICS session. When a session is traced, a screen image is captured in a trace data set whenever a TYPE command sends data to an online system. Another screen is captured when a response or other message sent by an online system is received at the virtual terminal used for a session.

In the following example, an exec named TRACE uses TRACE and ENDTRACE commands to capture a CICS session.

```
      /* Exec: TRACE VERIFY commands for REXX:       */
      /*       TRACE LOGON INVITE TYPE ENDTRACE      */
(1)     address TSO
       " allocate f(tracedd) sysout(a) "
(2)     address Verify
       " TRACE to tracedd image "
(3)    " LOGON cics1 model mod2 "
       " INVITE 1000 "
       " TYPE clear "
       " TYPE 'cemt set dataset(somedd) open' ",
       " enter "
         if substr(PTEBUFF,224,6) = 'NORMAL' then
            say 'open worked ok'
         else
            say 'open failed!!!'
       " TYPE pf3 "
       " TYPE home 'logoff' erase enter "
(4)    " ENDTRACE "
(5)     address TSO
       " free f(tracedd) "
```

Ports and Multiple Sessions

A CA Verify for CICS port is a place to anchor the application session, session trace, data queue and/or a REXX environment. Each CA Verify for CICS port is identified by a number. When the first exec in an address space issues a command, the exec's REXX environment is anchored to port 1.

CA Verify for CICS commands that operate on sessions, traces, execs or queues, create or use the session, trace, REXX environment or queue anchored to the numbered port being accessed by the exec when the command is issued. By default, an exec accesses the port to which its REXX environment is anchored. If an exec issues a CA Verify for CICS LOGON command, causing the exec's REXX environment to be anchored to port 1, the LOGON command allocates and anchors a virtual terminal to port 1 and uses the terminal to start a session on port 1.

While only one session may be active on a particular port at any point in time, an exec may use the ACCESS command to access another port and may use the LOGON command to start a session on the accessed port.

For example, ACCESS 2 instructs subsequent commands to create or use port 2 resources. While accessing port 2, a LOGON command would allocate a virtual terminal and use it to start a session on port 2, and TYPE commands would use the port 2 virtual terminal keyboard to enter transactions. An exec may use the ACCESS command multiple times to start multiple sessions on different ports, and may access a specific port at any time to enter transactions using the session anchored to the accessed port.

In addition to instructing subsequent commands to create or use resources anchored to a particular port, each ACCESS command also instructs session commands to return session information to an exec using either simple or compound session variable names.

- Simple variable names used by the application to return session information, including PTEBUFF, are defined in the application REXX Variables.
- Compound session variable names like PTEBUFF.1, or PTECOLS.3, have a REXX stem that is a defined simple session variable name and a tail that is the accessed port number.

By default, or if an exec issues an ACCESS EXECPORT command to access the port its REXX environment is anchored to, session commands return session information to the exec using the defined simple session variable names.

If an exec issues an ACCESS NEWPORT command to access a previously unused port, or an ACCESS FREEPORT command to reuse a port that is no longer being used, or if an exec issues an ACCESS command that specifies a port number, including the number of the port that the exec's environment is anchored to, the ACCESS command instructs subsequent session commands to return session information to the exec using compound session variable names.

As a result, when an exec uses the ACCESS command to start and use multiple sessions on multiple ports, the exec may refer to returned session information using either simple variable names or using compound variable names that have an actual or variable port number as a tail, that refers to information about a particular session.

The REXX exec in the following example uses ACCESS commands to start three sessions on three ports, to run three long running end-of-day transactions concurrently. The exec also uses CA Verify for CICS HANDLE and WAIT commands to control session flow and timing.

```
     /* Exec: MULTISES CA Verify commands for REXX: */
     /* ACCESS LOGON INVITE HANDLE WAIT */
        address Verify
(1)     sys.1 = 'cicsa'
        sys.2 = 'cicsb'
        sys.3 = 'cicsc'
        do port=1 to 3
       " ACCESS &port "
       " LOGON &sys.port model mod2 "
       " INVITE 1000 "
       " TYPE clear "
       " HANDLE turnaround off "
       " TYPE 'long running end-of-day tran' ",
       " enter "
       " HANDLE turnaround on "
        end
(2)    " WAIT 10000 on response "
        do until done = 3
           done = 0
           do port=1 to 3
              if PTEKB.port = 'INHIBITED' then do
       " ACCESS &port "
       " INVITE "
        end
       else
        done = done + 1
       end
    end
```

Sharing Sessions with Called Execs

When an exec calls another exec, the called exec runs in the same REXX environment as the calling exec, and by default accesses the port to which the REXX environment is anchored.

An exec may issue a LOGON command to start a session, then call another exec that issues a TYPE command to sign on. However, REXX variable pools are not shared by called and calling execs. When the called exec issues the TYPE command, PTEBUFF and other session variables are returned to the called exec, and PTEBUFF and other session information in the calling exec's variable pool becomes obsolete.

If an exec starts a session then calls another exec that uses the session, when the called exec returns, the calling exec may refresh session variables in its REXX variable pool using the QUERY SESSION command.

In the following example LOGON issues a LOGON command, then calls SIGNON to sign on. When SIGNON returns, LOGON uses the QUERY SESSION command to determine the status of the session and to refresh PTEBUFF and other session information.

```
/* Exec: LOGON Verify commands for REXX: */
/* LOGON QUERY TYPE */
   address Verify
 " LOGON cics1 model mod2 "
   call 'SIGNON'
 " QUERY session "
   select
      when PTEINFO = 'NOSESSION' then do
         say 'session failed'
         exit
         end
      when substr(PTEBUFF,2,15),
         \= 'Signon Complete' then do
         say 'sign on failed'
         exit
         end
      otherwise
 " TYPE clear "
      end
/* etc. */
/* Exec: SIGNON Verify commands for REXX: */
/* INVITE TYPE */
   address Verify
 " INVITE 1000 "
 " TYPE 'uid1' token 'uid1a' enter "
    return
```

CA Verify for CICS resources are cleaned up automatically when the first exec that runs in the port 1 REXX environment ends. If an exec called another exec to start a session before it issued a CA Verify for CICS command, the REXX environment would be anchored to port 1 when the called exec issued the LOGON command, and would be the first exec in the port 1 REXX environment. The session would be cleaned up automatically by CA Verify for CICS when the called exec returned to the calling exec.

If no active exec in an address space is using CA Verify for CICS commands, and an exec is executed that intends to use resources created by a called exec, it must issue some CA Verify for CICS command, for example, ACCESS EXECPORT, before calling an exec that creates the resources, to ensure that the resources are not cleaned up when the called exec ends.

Attaching Asynchronous Exec Tasks

When an exec creates multiple sessions on multiple ports, then uses the ACCESS command to switch from session to session, the events on the multiple sessions occur serially or synchronously. That is, only one TYPE command can be active at a time. Each TYPE command must complete before the exec can ACCESS another port or execute another TYPE command. The same is true if an exec calls another exec, because the called exec must return before the calling exec can execute additional session commands.

The CA Verify for CICS ATTACH command may be used to attach an exec that runs at the same time as the attaching exec. Unlike calling and called execs that run synchronously and in the same REXX Language Processor Environment, attaching and attached execs run asynchronously in different REXX Language Processor Environments and as separate z/OS tasks.

An ATTACH command specifies the name of the exec to be attached and may specify up to 20 literal or variable argument values to be passed to the attached exec. An ATTACH command may also specify REXX libraries to be used to load execs, the destination of REXX SAY messages issued by the attached exec and/or the destination of error messages issued by REXX while the exec is executing.

The ATTACH command creates a new z/OS task and a REXX Language Processor Environment that are used to run the attached exec. The REXX environment is anchored to the port being accessed by the attaching exec when the command is issued. Before issuing an ATTACH command, an ACCESS command must be issued to access a CA Verify for CICS port that is not already associated with a REXX environment.

After loading the attached exec in the new REXX environment, the ATTACH command completes and the attaching exec resumes execution at the same time that the attached exec begins execution.

Attached execs run as peer-to-peer tasks and are not arranged in a hierarchy. No notification is given to an attaching task when an attached task ends. An attaching exec may end before an attached exec ends. Any exec in an address space may use the ATTACH command to attach any other exec and any attached exec may use any CA Verify for CICS command to create application resources or to use resources created by any other exec running on any port in the address space.

Sessions, traces, data queues and other CA Verify for CICS resources created by execs are cleaned up automatically by CA Verify for CICS only when the first exec running in the port 1 REXX environment end. When the port 1 exec ends, cleanup is delayed until any other active execs that have issued CA Verify for CICS commands have ended.

In the following example an exec named FAST, attaches another exec named CHANGE 10 times to change a user password in 10 remote systems. The result would be the same if FAST called CHANGE10 times instead of attaching it. However, because called execs run one at a time, and because attached execs run at the same time, the ATTACH solution runs 10 times faster.

```
     /* Exec: FAST Verify commands for REXX: */
     /* ACCESS ATTACH */
(1)     arg uid oldpt newpt
        address Verify
        do n=1 to 10
(2)    " ACCESS newport "
       " ATTACH exec change ",
       " &uid &oldpt &newpt 'tso"n"' "
            end
     /* Exec: CHANGE Verify commands for REXX: */
     /* MONITOR LOGON INVITE TYPE */
(3)     arg uid oldpt newpt tson
        address Verify
       " MONITOR signals "
(4)    " LOGON &tson data &uid "
        do until substr(PTEBUFF,34,11),
           = 'TSO/E LOGON'
       " INVITE 1000 "
           end
(5)    " TYPE token &oldpt tab token &newpt enter "
        if substr(PTEBUFF,11,16)
           ª= 'REENTER PASSWORD' then do
       " TYPE token token &newpt enter "
          if substr(PTEBUFF,11,16)
          ª= 'PASSWORD CHANGED' then do
              say tson 'password changed'
          exit
             end
          end
        say tson 'password change failed'
```

Queues and Intertask Communication

Attached execs that run as asynchronous z/OS tasks may use CA Verify for CICS data queues to communicate. Like the commands that attach execs and manage sessions, CA Verify for CICS commands that manage data queues operate on the data queue anchored to the port being accessed by an exec when a command is executed.

QUEUE, PUSH, and POST Commands

QUEUE, PUSH and POST commands may store up to 20 literal or variable argument values as a single entry in an accessed queue. PULL and QUERY QUEUE commands return the argument values fetched from an accessed queue entry in corresponding REXX variables specified by the PULL or QUERY QUEUE command.

- The QUEUE command adds an entry to the bottom of a queue.
- The PUSH command adds an entry to the top of a queue.

- The POST command replaces an entire queue with a specified queue entry.
- The PULL command returns data values fetched from the top entry in a queue and deletes the entry from the queue.
- The QUERY QUEUE command returns data values fetched from the top entry in a queue, but leaves the entry in the queue.
- The DELETEQ command deletes an entire data queue.

Data queues can be used in any desired way:

- A queue may be used by a single exec as private temporary storage.
- A queue may be used as a bulletin board to post information that may be examined by other execs.
- A queue may be used to collect data that is stored by one or more execs and is processed by a single exec.
- A queue may be used to distribute data to multiple execs that access a common queue.
- Multiple queues may be used to distribute data to execs that access designated queues.

By default, if an accessed queue is empty when a PULL command is issued, RC=12is returned to the exec and the command completes immediately. An exec designed to process data queued by other execs may use the WAIT command to instruct subsequent PULL commands to wait a specified amount of time or to wait FOREVER for an entry to be queued, if the accessed queue is empty when a PULL command is issued.

The SERVER and SESSION execs in the following examples illustrate two uses of CA Verify for CICS data queues. The SERVER exec attaches SESSION execs to multiple ports and distributes a file of data to the queues anchored to SESSION exec ports. Each SESSION exec starts a session and uses the session to process queue entries pulled from the queue anchored to the SESSION exec's port. After processing each distributed queue entry, the SESSION exec returns a status message to the queue anchored to the SERVER exec's port. After logging a status message pulled from the SERVER exec's queue, the SERVER exec distributes another record to the SESSION exec that returned the logged status message.

Attached execs that drive asynchronous sessions and communicate using data queues may be used to perform system stress testing or may be used in production applications to multiply the throughput possible using conventional serial data processing methods.

```
     /* Exec: SERVER Verify commands for REXX: */
     /* ACCESS QUEUE ATTACH WAIT PULL */
(1)     address TSO
       " alloc f(quefile) da('appl.quedata') shr "
       " alloc f(logfile) da('appl.logdata') mod "
        querecs = 0
(2)     do 10
address TSO
     " execio 1 diskr quefile      "
if rc \= 0 then do
querec = 'eof'
leave
end
pull querec
querecs = querecs + 1
address Verify
     " ACCESS newport      "
     " QUEUE &querec      "
     " ATTACH exec session &pteport      "
end
address Verify
     " WAIT forever on pull      "
(3) do while querec \= 'eof'
     " ACCESS execport      "
     " PULL &logrec &port      "
address TSO
queue logrec
     " execio 1 diskw logfile      "
querecs = querecs - 1
     " execio 1 diskr quefile      "
if rc = 0 then do
pull querec
querecs = querecs + 1
end
else
querec = 'eof'
address Verify
     " ACCESS &port      "
     " QUEUE &querec      "
end
(4) do while querecs > 0
     " ACCESS EXECPORT      "
     " PULL &logrec &port      "
address TSO
queue logrec
     " execio 1 diskw logfile      "
querecs = querecs - 1
address Verify
          " ACCESS &port           "
          " QUEUE 'eof'           "
end
(5) address TSO
          " execio 0 diskr quefile (FINIS           "
          " free f(quefile)           "
          " execio 0 diskw logfile (FINIS           "
          " free f(logfile)
```

SERVER exec notes:

1. The SERVER exec allocates a quefile that contains data to be processed, allocates a logfile used to collect status messages and initializes a count of in-progress quefile records to zero.

2. The SERVER exec startup loop uses EXECIO and the REXX PULL instruction to read records sequentially from the quefile, and counts each record read as an in-progress quefile record. For each record read, the CA Verify for CICS ACCESS command is used to access an unused port, the QUEUE command is used to store the quefile record in the data queue anchored to the accessed port, and an ATTACH command is used to attach the SESSION exec to the accessed port to process quefile records distributed to the SESSION exec by the SERVER exec. An ATTACH parameter provides the SERVER exec’s port number to each attached SESSION exec.

   SESSION execs use the SERVER exec's port number to return status messages to the SERVER exec after processing each distributed quefile record. Before processing status messages, the SERVER exec uses the CA Verify for CICS WAIT command to instruct subsequent PULL commands to wait for a status message to be queued if the SERVER exec’s queue is empty when a PULL command is issued.

3. Each iteration of the SERVER exec mainline loop logs a status message received from any attached SESSION exec and supplies another quefile record to the SESSION exec. The SERVER exec uses the ACCESS command to access the SERVER exec's port and uses the CA Verify for CICS PULL command to wait for and pull an entry from the SERVER exec’s queue. Each queue entry added to the SERVER exec's queue by an attached SESSION exec contains a status message and also the port number of the SESSION exec that added the entry to the SERVER exec’s queue. The REXX QUEUE instruction and EXECIO are used to write each pulled status message to the logfile managed by the SERVER exec, and the count of in-progress quefile records is decremented after logging the status message for a processed quefile record.

   After logging a status message returned by a SESSION exec, the SERVER uses EXECIO and the REXX PULL instruction to read another quefile record to be processed by the SESSION exec, and counts each record read as an in-progress quefile record. When the end of the quefile is reached, an 'eof' record is created to inform the SESSION exec that all quefile records have been processed and to signal the end of the SERVER mainline loop. The SERVER exec uses the port number that was pulled with the previous status message to ACCESS the SESSION exec's port and uses the QUEUE command to store either the next quefile record or the eof record in the SESSION exec's queue.

4. When the end of the quefile has been reached, the SERVER exec pulls, logs and accounts for the status message for each in-progress quefile record, and queues an eof record to each SESSION exec that returns a final status message, until the count of in-progress quefile records is zero.
5. After all quefile records have been processed and all status messages have been logged, quefile and logfile are closed and freed, and the SERVER exec ends.

```
/* Exec: SESSION Verify commands for REXX: */
/* LOGON INVITE TYPE LOGOFF */
/* WAIT ACCESS PULL QUEUE */
(1) arg logport
address Verify
" LOGON 'cics1' mod2 "
" INVITE 1000 "
" TYPE 'uid' token 'uida' enter "
" TYPE clear "
" TYPE 'tran' enter "
" WAIT forever on pull "
(2) do forever
" ACCESS execport "
" PULL &querec "
if querec = 'eof' then
leave
" TYPE &querec enter "
logrec = substr(PTEBUFF,23*80+1,80)
" ACCESS &logport "
" QUEUE &logrec &pteport "
end
(3) " TYPE clear "
" LOGOFF
```

SESSION exec notes:

1. Each attached copy of the SESSION exec uses the REXX ARG instruction to get the SERVER exec port number passed to the SESSION exec as a parameter by the ATTACH command used by the SERVER exec to attach the SESSION exec.

   The SESSION exec starts a session, signs on, and enters a tran that could be any transaction that processes any data contained in any querec.

   The CA Verify for CICS WAIT command is used to instruct subsequent PULL commands to wait for a querec to be queued if the SESSION exec's queue is empty when a PULL command is issued.
2. Each iteration of the SESSION exec mainline uses ACCESS and PULL commands to wait for and to pull a querec from the SESSION exec's queue, uses CA Verify for CICS session commands to enter the pulled querec, and uses ACCESS and QUEUE commands to store an entry in the SERVER exec’s queue. The queue entry includes a status message displayed by 'tran' on screen row 24, and also includes the port number of the SESSION exec that stored the queue entry.
3. An eofquerec instructs the SESSION exec to clean up its session and end.

Sharing Resources with Attached Execs

Attached execs run as z/OS tasks and may be executing at the same time using multiple hardware processors. When attached execs share CA Verify for CICS resources and multiple execs attempt to use a shared resource at exactly the same time, processor access to the shared resource is serialized automatically. For example, if two execs attempt to pull data from the same queue at the same time, the requests are processed serially and each exec pulls a different entry from the queue.

Locks are used to serialize access to resources. When an exec issues a CA Verify for CICS command that uses a shared resource, a lock is obtained that permits exclusive use of the resource by the exec task that owns the lock, before the resource is used. If an exec issues a command that uses a shared resource, while the resource lock is owned by another task, the task that wants the lock queues a lock requests and waits for the lock.

After a requested operation has been performed on a locked resource, if a lock request was queued by another exec task, ownership of the resource lock is transferred to the task that queued the oldest lock request and execution of the task that then owns the lock is resumed, otherwise the resource is unlocked.

Resource locks are obtained automatically by CA Verify for CICS commands when shared resources are used, but are held only for the life of the command. In some applications it may be necessary or convenient to hold resource locks while issuing a series of commands. An exec may use the CA Verify for CICS LOCK command or an ACCESS command that specifies the LOCK keyword to lock an accessed port. When a port is locked, the port and all resources anchored to the port are locked. A port lock is held until an UNLOCK command is issued.

If an exec issues commands that use resources anchored to a port that is locked by another exec task, or attempts to lock a port that is locked by another exec task, lock requests are queued and remain queued until the exec that holds the port lock unlocks the port.

An exec may lock a port to issue a series of commands that use resources anchored to the port, while preventing other execs from using port resources while the commands are executed. For example:

- An ACCESS FREEPORT LOCK command accesses a port that is devoid of resources and locks the port until an UNLOCK command is issued. Two or more execs may issue the command at the same time to attach other execs to the accessed ports without the possibility that the ACCESS command would assign the same FREEPORT to two execs before either had a chance to issue the ATTACH command.
- If an exec task is used to log records that are queued by multiple other exec tasks, and it is desirable to log a series of related records that are queued by a single exec task without interleaving the records with records queued by other exec tasks, LOCK and UNLOCK commands may be used by the execs that queue records, to prevent other exec tasks from queuing records while a series of related records are being queued.

Smart Exec-Driven Sessions

REXX execs created by converting logged test streams contain CA Verify for CICS LOGON, TYPE, INVITE and DELAY commands that start sessions, re-key user keystrokes, and that handle other detected session events. When they are executed, converted REXX execs create sessions that are similar to the logged user session, provided that online system responses to re-keyed input are similar to online system responses to original input.

While converted REXX execs repeat only what a person did, and do not know why they did it or what they would have done if a system had responded to input in a different way, when a converted exec is executed, the virtual terminal screen images returned by session commands in REXX variables provides the same information to an exec that displayed online system responses provide to a terminal user.

REXX instructions and functions, returned session information and other CA Verify for CICS commands and command options may be used to develop intelligent REXX execs that make the decisions that people do. For example:

- REXX execs may examine virtual terminal screen images to verify that transactions completed successfully and to decide what to do next.
- Information from previous online system responses may be saved and used as input to subsequent transactions.
- The CA Verify for CICS QUERY command may be used to check the status of or request information about CA Verify for CICS resources.

- The CA Verify for CICS SIGNAL command may be used to trap session failures or other unusual events using REXX condition traps or return codes may be checked to detect the events.
- The CA Verify for CICS WAIT command may be used to interrupt session commands if an online system response is not received in a reasonable amount of time.

Intelligent REXX applications that use CA Verify for CICS session commands and any other resources available to REXX execs may be used to perform or automate almost any imaginable terminal task that could be performed by a person using a terminal.

ISPF Session Panels

Options of the CA Verify for CICS LOGON and TYPE commands make it easy to display CA Verify for CICS virtual terminal screen images using an ISPF panel and to rekey data entered at the displayed panel into the virtual terminal display.

The LOGON command PANELDATA keyword instructs subsequent CA Verify for CICS session commands to edit the virtual terminal display buffer returned in variable, PTEBUFF, for display using an ISPF panel similar to the panel shown in the following example.

```
)ATTR
01 TYPE(DATAIN)  INTENS(LOW)  SKIP(OFF) CAPS(OFF) JUST(ASIS)
05 TYPE(DATAIN)  INTENS(HIGH) SKIP(OFF) CAPS(OFF) JUST(ASIS)
07 TYPE(DATAIN)  INTENS(NON)  SKIP(OFF) CAPS(OFF) JUST(ASIS)
11 TYPE(DATAOUT) INTENS(LOW)  SKIP(OFF) CAPS(OFF) JUST(ASIS)
15 TYPE(DATAOUT) INTENS(HIGH) SKIP(OFF) CAPS(OFF) JUST(ASIS)
19 TYPE(DATAOUT) INTENS(LOW)  SKIP(ON)  CAPS(OFF) JUST(ASIS)
1D TYPE(DATAOUT) INTENS(HIGH) SKIP(ON)  CAPS(OFF) JUST(ASIS)
| AREA(DYNAMIC)  EXTEND(ON)   USERMOD(3F)
)BODY WIDTH(80)
|PTEBUFF |
)INIT
.ALARM = &PTEALARM
)PROC
&CSRPOS = .CSRPOS
&PFKEY = .PFKEY
)END
```

1. When a LOGON command specifies PANELDATA, basic 3270 attributes in the virtual terminal display buffer are returned in PTEBUFF as one of the seven attributes in the session panel definition. 3270 Attribute Values documents the mapping of basic 3270 attributes to panel attributes.
2. PANELDATA also edits PTEBUFF data to remove nulls in protected fields that would otherwise be displayed as periods by ISPF.

3. When the session panel is used to display a virtual terminal screen image, ISPF displays the PTEBUFF variable value returned by CA Verify for CICS session commands, in the PTEBUFF dynamic area defined in the body of the session panel.
4. The USERMOD(3F) specification in the panel definition instructs ISPF to replace a field attribute in the PTEBUFF variable with the hexadecimal value, 3F, when data is keyed into a field. In a CA Verify for CICS LOGON command, USERMOD '3F' informs CA Verify for CICS of the USERMOD value defined in a session panel.
5. CA Verify for CICS session commands return YES in variable, PTEALARM, when a message received from an online system sounds the terminal alarm. The reference to PTEALARM in the panel definition causes ISPF to echo an alarm at the terminal if an alarm was sounded at the virtual terminal.
6. The PROC section of the panel definition instructs ISPF to return the cursor position and the PF key in variable fields CSRPOS and PFKEY, when data is entered and the ISPF DISPLAY command completes.

The exec named SESSMAN, shown in the following example is a TSO command that starts and manages an interactive session with a named online system. For example, 'TSO SESSMAN CICSA', may be entered while using any ISPF panel to start a session with an online system, CICSA. The SESSMAN exec uses a session panel, the ISPF DISPLAY command, and CA Verify for CICS session commands to start and display a virtual terminal session at a real terminal, rekey data entered at a terminal into the virtual terminal display, and send the data to an online system. To the user, a session managed by SESSMAN works much like any other terminal session.

```
    /* Exec: SESSMAN Verify commands for REXX: *    /
    /* MONITOR SIGNAL LOGON DELAY TYPE *    /
(1) arg system
address Verify
(2)      " monitor signals      "
     " signal error on logoff      "
(3)      " logon &system paneldata usermod '3F'      ",
     " 24x80 24x80 basicds      "
(4)      " delay 500      "
     " signal off on logoff      "
do forever
address ispexec ,
(4)      " display panel(sesspan)      ",
     " cursor(ptebuff) csrpos(     "ptecsrp     ")      "
(5) if pfkey = ' ' then
aid = 'enter'
else
aid = pfkey
     " type usermod <&csrpos> aid &aid      "
(6) if rc     /= 0 then
leave
end
```

SESSMAN Exec Notes:

1. The name of an online system specified in a SESSMAN command is received by the SESSMAN exec as a REXX argument.
2. CA Verify for CICS MONITOR and SIGNAL commands are used to display an error message if a requested session cannot be started.
3. A CA Verify for CICS LOGON command is used to start a session with the named online system. The command specifies PANELDATA to instruct subsequent session commands to edit PTEBUFF for display using a session panel, and specifies USERMOD '3F' to inform CA Verify for CICS of the USERMOD value defined by the session panel.
4. After the initial sign on screen is received by the DELAY command, the SESSMAN exec enters a DO loop that ends only when the session ends.
5. An image of each display received at the virtual terminal is displayed at the TSO terminal using the ISPF DISPLAY command and a session panel like the panel shown in the following example. The DISPLAY command positions the cursor using the virtual terminal cursor position returned by session commands in variable, PTECSRP.
6. A TYPE command that specifies the USERMOD keyword is used to rekey data entered by the terminal user into the virtual terminal display. USERMOD finds data in PTEBUFF that was entered by the user using the USERMOD value specified by the LOGON command and in the session panel definition. The TYPE command uses the ENTER key or the PF key pressed by the terminal user to send the input to the online system.
7. The session and the SESSMAN exec end when the user enters a logoff transaction that causes the online system to end the session.

Monitoring Execs and Sessions

CA Verify for CICS commands are designed to automate terminal tasks and provide options that allow REXX execs to handle all possible errors and other events, and therefore, by default, do not automatically display error messages, exec-driven sessions or other command events. However, an exec or an exec user may use command monitoring functions built into CA Verify for CICS commands for REXX, to monitor selected events and to document, display or otherwise process errors, exec-driven sessions and other events automatically, while an exec is executing.

The CA Verify for CICS MONITOR command may be used in any exec to automatically display information about specified events while an exec is executing or to call a monitor exec to process the information:

- MONITOR SIGNALS detects REXX conditions raises by CA Verify for CICS commands and may be added to any exec to display messages automatically when REXX conditions are raised.
- MONITOR CONDITIONS detects non-zero return codes returned by CA Verify for CICS commands and may be used to display messages that document the reason.
- MONITOR COMMANDS may be used to display CA Verify for CICS commands as they are issued by REXX execs.
- MONITOR RESPONSES calls a default or specified monitor exec when responses or other outbound session events are received and may be used to display an exec-driven session at a terminal.
- MONITOR STORE, DROP, FETCH and/or one or more generic variable names, specified in any combination, may be used to display variable values when they are changed, used or dropped by CA Verify for CICS commands.

MONITOR command VIASAY, VIATPUT and VIAEXEC options specify how monitored events are handled when they are detected. VIASAY and VIATPUT display events other than RESPONSES using the REXX SAY instruction or TSO TPUT, depending on the REXX environment in which the exec is executing. VIAEXEC calls a specified monitor exec when a monitored event is detected. When RESPONSES are monitored, a monitor exec is always called to process online system responses received by CA Verify for CICS session commands, either an exec specified by the VIAEXEC option or a default monitor exec.

When a monitor exec is called to process detected events, call arguments provide detailed information about the event to the monitor exec. Call arguments are explained in The Monitor Exec Interface. The monitor exec shown in The Distributed Monitor Exec VTEMONX is designed to be used as the default monitor exec that is called when RESPONSES are monitored but a VIAEXEC is not specified. When it is called to process an online system response to a transaction entered by an exec using the CA Verify for CICS TYPE command, VTEMONX uses either ISPF display services or the REXX SAY instruction to display a virtual terminal screen image, depending on the REXX environment in which the exec is executing.

A MONITOR command applies to all execs that execute in the REXX environment in which the MONITOR command is executed, and may be issued by a general purpose front-end exec that is used to call and monitor other execs with no modification to the called execs.

The distributed front-end exec, VTESMON, is shown in the following example. VTESMON may be used to call and monitor an exec in REXX environments created by ISPF, TSO or IRXJCL, with no modification to the called exec, by entering the following as a TSO command or as an IRXJCL PARM:

```
VTESMON testexec args
```

VTESMON issues a MONITOR command to monitor CA Verify for CICS commands issued by the called exec, testexec, relying on the default monitor exec, VTEMONX, to display response events. Both VTESMON and VTEMONX use either ISPF display services or the REXX SAY instruction to display events depending on the REXX environment. When a specified testexec returns, VTESMON reports any REXX condition caused by the call to testexec and ends any session not ended by testexec.

The following is an example of the VTESMON exec-driven session monitor:

```
   /* VTESMON - A Monitor for REXX-driven Verify Sessions (distributed) */
       arg args
       parse upper var args execname execargs
       if execname = '' then do
           say 'Enter: VTESMON execname (optional args for execname)'
           exit
           end
       address Verify
     " ACCESS EXECPORT " ; " QUERY SESSION "
       if ptestate \= 'PTESTATE' then " LOGOFF "
     " MONITOR SIGNALS RESPONSES "
         signal on halt name halt     ; signal on error name error
         signal on syntax name syntax ; signal on failure name fail
         interpret "call '"execname"' '"execargs"'"
         problem = ''                 ; signal finish
    halt: problem = 'HALT'            ; signal finish
   error: problem = 'ERROR'           ; signal finish
  syntax: problem = 'SYNTAX'          ; signal finish
    fail: problem = 'FAILURE'         ; signal finish
  finish: signal off halt             ; signal off error
         signal off syntax            ; signal off failure
    " MONITOR OFF " ; " ACCESS LASTPORT " ; " QUERY SESSION "
    parse source . . . . . . addrspn .
    if addrspn = 'ISPF' then do
          green = '01'x ; red = '05'x ; blue = '11'x ; white = '15'x
          PTEDESKT = ''
          address ispexec
          " control nondispl " ; " display panel(VTEPXMDT) "
          zwinttl = execname 'execution ended on Port' PTEPORT
          select
               when problem \= '' then
                  PTEMWIN2 = left(red || execname || white,
                                  || 'terminated; REXX condition(',
                                  || red || problem || white || ').',72),
                                  || right(blue || 'Press' || green,
                                  || 'ENTER' || blue || 'to end test.',72)
          when ptestate.pteaport \= 'PTESTATE.'PTEAPORT then
                   PTEMWIN2 = left(red || execname,
                                  || white || 'did not end Port' PTEAPORT,
         'session. ' red'LOGOFF issued.',72),
                                  || right(blue || 'Press' || green,
                                  || 'ENTER' || blue || 'to end test.',72)
          otherwise do
                                    PTEMWIN2 = left(green || execname,
                                      || white || 'ended normally.',72),
                                      || right(blue || 'Test ending . . .',72)
          " control display lock "
               end
             end
          " addpop row(19) column(1) " ; " display panel(VTEPXMW2) "
          address Verify " WAIT 800 "
              end
        else do
           say copies('_',79) ; say ' '
           select
               when problem \= '' then
                     say execname 'terminated; REXX condition(' problem ').'
               when ptestate.pteaport \= 'PTESTATE.'PTEAPORT then
                     say execname 'ended but did not end Port',
                              PTEAPORT 'session. LOGOFF issued.'
                     otherwise say execname 'ended normally.'
                     end
```

```
            end
        if ptestate.pteaport \= 'PTESTATE.'pteaport then " LOGOFF "
```

Chapter 13: REXX Reference

This chapter documents the function and syntax of each host command used in the CA Verify for CICS Host Command Environment. The commands are presented in alphabetical order according to command name.

This section contains the following topics:

Syntax Notation Format

- CA Verify for CICS Host Commands and keywords are in UPPERCASE text.
- User data is listed in lowercase bold text and can be literal data or variables.
- Optional parameters are enclosed in parentheses.
- Alternative keywords and/or parameters are separated by the bar symbol.
- In CA Verify for CICS commands, REXX variable names are coded with a leading ampersand, &, to distinguish them from keywords and other command operands.
- Host command continuation is indicated with a comma.

ACCESS

When CA Verify for CICS commands start or use sessions, start or end session traces, queue or pull queued data or perform other command functions, by default, they anchor or use resources anchored to the CA Verify for CICS port on which the issuing exec is executing, and session commands return session information using the simple session variable names defined in CA Verify for CICS REXX Variables.

The command:

- Instructs subsequent CA Verify for CICS commands to anchor or use resources anchored to a specified port when performing a command function.
- Instructs subsequent session commands to return session information using either simple or compound session variable names.
- Returns the accessed port number in REXX variable, PTEAPORT.
- Returns the current exec port number in PTEPORT.

Compound session variable names, like PTEBUFF.1, have a REXX stem that is a defined simple session variable name and a tail that is the accessed port number.

Examples:

```
" ACCESS EXECPORT "
" ACCESS NEWPORT  "
" ACCESS 3 "
" ACCESS &port "
```

Format:

```
ACCESS ( EXECPORT | * | port |  NEWPORT | FREEPORT | LASTPORT )  ( LOCK )
```

ACCESS EXECPORT or ACCESS *

Instructs subsequent commands to anchor or use resources anchored to the port on which the exec is executing, and instructs session commands to return information using simple session variable names.

ACCESS port

Where port is either a port number or a REXX variable name, coded with a leading ampersand, that contains a port number, instructs subsequent commands to anchor or use resources anchored to the specified port, and instructs session commands to return information using compound session variable names.

ACCESS NEWPORT(the default)

Assigns an unused port number, instructs subsequent commands to anchor or use resources anchored to the assigned port, and instructs session commands to return information using compound session variable names.

ACCESS FREEPORT

Locates a port on which no exec, session, session trace, data queue or lock is active, or assigns an unused port number, instructs subsequent commands to anchor or use resources anchored to the located or assigned port, and instructs session commands to return information using compound session variable names.

ACCESS LASTPORT

Identifies the port that was last accessed by any exec running on the current exec port, instructs subsequent commands to anchor or use resources anchored to the identified port, and instructs session commands to return information using compound session variable names.

LOCK

Can be specified to lock the accessed port. When a port is locked, an UNLOCK command must be issued before port resources can be created or used by an exec running on any port other than the port that holds the lock.

Conditions and Return Codes

At the completion of an ACCESS command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes ACCESS command return codes and default REXX conditions that may be raised for some conditions.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
CANCEL	24	HALT	Cancel command processed
FAULT	64	HALT	Invalid command or system error

ATTACH

The ATTACH command creates a new z/OS task and REXX Language Processor environment, anchors the environment to the CA Verify for CICS port being accessed by the exec when the ATTACH command is issued, and executes a specified exec in the new REXX Language Processor Environment.

The difference between calling an exec and attaching an exec is that the REXX CALL instruction completes and the calling exec resumes execution only when the called exec returns to the calling exec. However, the ATTACH command completes and the attaching exec resumes execution at the same time that the attached exec begins execution. Called and calling execs run one at a time, while attached and attaching execs run at the same time.

Examples:

```
" ATTACH EXEC &execname    "," &arg1 'arg2' "

" ATTACH EXEC subexec 'logon' "" &sysid &uid &code  "
```

Format:

```
ATTACH EXEC execname ( arg1 . . . arg20 )  ( EXECDD execdd )  ( SAYDD saydd )  ( SAYDEST
saydest )  ( MSGDEST msgdest )
```

EXEC

A required keyword which indicates that the value that follows is the name of the exec to be attached.

execname

The name of the REXX exec to be attached. It may be the actual exec name, a literal whose value is the exec name, or a variable name coded with a leading ampersand that contains the exec name (for example, &execname).

arg1 through arg20

Up to 20 arguments to be passed to the attached exec. Each argument may be a literal value enclosed in single or double quotes, or may be a variable name, coded with a leading ampersand that contains an argument value (for example, &arg).

Note: An attached exec may use a REXX ARG instruction to assign argument values to corresponding variables specified by REXX ARG instruction. ARG instruction variable names are separated by commas.

EXECDD

An optional keyword which indicates that the value that follows is the DD name of the REXX library(s) from which the attached exec, and other execs, are to be loaded into the REXX Language Processor Environment created by the ATTACH command.

execdd

The exec library DD name from which execs are loaded. It may be the actual DD name, a literal whose value is the DD name, or a variable name coded with a leading ampersand that contains the exec library DD name (for example, &execdd). If an EXECDD is not specified, execs are loaded from the exec libraries used in the first REXX Language Processor Environment in an address space that uses CA Verify for CICS commands for REXX, usually from SYSEXEC and/or SYSPROC.

SAYDD

An optional keyword which indicates that the value that follows is the DD name to which REXX SAY messages are to be written in the REXX Language Processor Environment created by the ATTACH command. The SAYDD keyword option is subject to system options, described below, that control REXX message management options.

saydd

The DD name to which REXX SAY messages are to be written. It may be the actual DD name, a literal whose value is the DD name, or a variable name coded with a leading ampersand that contains the DD name (for example, &saydd).

A null-valued variable may be specified as a saydd to request allocation of a new message dataset using a unique system-assigned DD name, and that the assigned name is to be returned to the attaching exec in the specified saydd variable.

If a specified saydd is already allocated when an ATTACH command is executed, the allocated dataset is used in the new REXX Language Processor Environment as a message dataset. If a saydd that was not allocated by CA Verify for CICS is specified for two or more attached execs that run concurrently, the user must insure that the allocated dataset can be used as a shared message destination by the REXX Language Processor Environments created to run the attached execs.

SAYDEST

An optional keyword which indicates that the value that follows is to be allocated as the destination of REXX SAY messages, if the specified or default SAYDD is not allocated when the ATTACH command is executed. The SAYDEST keyword option is subject to system options, described below, that control REXX message management options.

saydest

The SAYDEST destination value. It may be a JES SYSOUT class, an asterisk, *, to specify the default SYSOUT class for the address space, or may be the keyword, TERMINAL to send REXX SAY messages to a TSO terminal. saydest may be the actual JES SYSOUT class or the TERMINAL keyword, may be a literal whose value is the SAY message destination, or may be variable name coded with a leading ampersand that contains the SYSOUT class or TERMINAL keyword (for example, &saydest).

MSGDEST

An optional keyword which indicates that the value that follows is the destination of REXX error messages issued while processing execs in the REXX Language Processor Environment created by the ATTACH command. The MSGDEST keyword option is subject to system options, described below, that control REXX message management options.

msgdest

A REXX error message destination identified by the MSGDEST keyword. Specify one of the following destination keywords:

- WTO to write REXX messages using WTO only
- SAY to write REXX messages to the REXX SAY message destination only
- ALL to write REXX messages using both WTO and to the REXX SAY message destination
- OFF to discard REXX error messages

Msgdest may be a destination keyword, may be a literal whose value is a destination keyword, or may be a variable name coded with a leading ampersand that contains a REXX message destination keyword (for example, &msgdest).

Note: The number of REXX execs that may be attached is limited by the REXX Language Processor Environment limit contained in the TSO/E REXX environment table, IRXANCHR. IBM publication, TSO/E REXX/z/OS Reference, explains how to change the REXX environment limit.

All REXX Language Processor Environments in which CA Verify for CICS commands are used are anchored to a CA Verify for CICS port. When commands are used by an exec running in a REXX environment not created by CA Verify for CICS, the environment is anchored to a port when the first command is issued. Only one REXX Language Processor Environment may be anchored to a port. The ATTACH command creates a new REXX Language Processor Environment and anchors the environment to the port being accessed by an exec anchored to the port. Before using the ATTACH command, an exec must use the CA Verify for CICS ACCESS command to access a NEWPORT, a FREEPORT, or to access a specific port that is not already associated with a REXX Language Processor Environment.

An attached exec may ADDRESS VERIFY and issue commands for REXX.

REXX Language Processor Environments created by the ATTACH command are not integrated into TSO. Attached execs may not use TSO and ISPF commands that are supported only in REXX Language Processor Environments that are integrated into TSO. However, execs that operate correctly when attached in a TSO address space may also be attached and operate correctly when IRXJCL is used to execute an exec as a batch job or when they are attached in other non-TSO address spaces.

Attaching Asynchronous Exec Tasks (see page 282) and Queues and Intertask Communication (see page 283) describe applications of the ATTACH command and other CA Verify for CICS commands designed for use by execs that run in a multi-task environment.

REXX Messages Management Options

The operation of and default values of the SAYDD, SAYDEST and MSGDEST keyword options are subject to CA Verify for CICS system options that allow an installation to specify REXX message management options for each named address space in which CA Verify for CICS commands for REXX are used. The name of an address space is defined by the REXX parameters module used to initialize REXX Language Processor Environments.

- System options may instruct CA Verify for CICS to ignore ATTACH command SAYDEST and SAYDD specifications in address spaces that manage REXX message destinations for all REXX Language Processor Environments in the address space.
- System options may instruct CA Verify for CICS to manage REXX message destinations only when an ATTACH command specifies SAYDEST and/or SAYDD options.

- System options may instruct CA Verify for CICS to ignore any MSGDEST specified by an ATTACH command, and to use a specified option when running in a named address space.
- If system options permit, CA Verify for CICS allocates and manages REXX message destinations and options as specified by ATTACH command SAYDEST, SAYDD and/or MSGDEST options or allocates and manages REXX message destinations using default values when ATTACH command options are not specified. Message datasets allocated by CA Verify for CICS are automatically freed when the last attached exec using a message dataset ends.

Conditions and Return Codes

At the completion of an ATTACH command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes ATTACH command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
LOCKOUT	4		Timed out waiting for port lock
CANCEL	24	HALT	CANCEL command processed
FAULT	64	HALT	Invalid command or system error

CANCEL

The CANCEL command causes the REXX HALT condition to be raised for the exec that is active on the port being accessed by the issuing exec.

Example:

```
" CANCEL      "
```

Format:

```
CANCEL
```

When the CANCEL command is executed, if an exec is active on the port being accessed, then cancellation of the exec is scheduled. The REXX HALT condition is subsequently raised when the cancelled exec recognizes the request following the execution of any CA Verify for CICS command.

When the CANCEL command is executed, if a CA Verify for CICS command is currently executing within the cancelled exec, and if the command is waiting for the completion of an event that is under CA Verify for CICS control, then the wait for the completion of the event is interrupted immediately. The REXX HALT condition is raised when the interrupted command ends.

If an exec is not active on the port being accessed when a CANCEL command is executed, then no action is performed, but the command is considered to have completed without error.

Conditions and Return Codes

At the completion of a CANCEL command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes CANCEL command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
CANCEL	24	HALT	CANCEL command processed
FAULT	64	HALT	Invalid command or system error

DELAY

The DELAY command delays execution of an exec for a specified amount of time, but allows outbound session traffic to flow on the session anchored to the port being accessed by the exec, during the delay.

Examples:

" DELAY 3200 "" DELAY PERCENT 50 "

Format:

DELAY hhmmssth | (PERCENT percent)

Hhmmssth

A delay time interval, expressed in hours (hh), minutes (mm), seconds (ss) and tenths and hundredths of seconds (th). A delay time interval may be specified using a REXX variable, coded with a leading ampersand, that contains the interval in the format hhmmssth.

PERCENT

An optional keyword, which specifies that delay times specified by DELAY commands subsequently executed by execs that run on the CA Verify for CICS port on which a DELAY PERCENT command is executed, are to be factored by the specified percent.

percent

May be an integer value from zero to 999, or may be a variable field that contains an integer from zero to 999. It is the percent of specified delay times to be used when DELAY commands are subsequently executed.

DELAY commands created by recording a session as a REXX exec capture the elapsed time, often called think time, that it took for a user to press enter or another key that sends data, after receiving the preceding message from the online system. When a DELAY command is executed, exec execution is delayed until the recorded or specified delay time elapses, simulating the think time delays that occur when people use terminals, and causing a session produced by a recorded exec to proceed at the same speed as the captured session.

The speed of execs that issue DELAY commands may be controlled using a DELAY command that specifies a delay PERCENT. When a DELAY PERCENT command is executed that specifies a percent less than 100, exec execution is delayed only for a percent of the time specified by each subsequent DELAY command and the exec runs faster. When a percent greater than 100 is specified, delays are longer than the delay times specified by subsequent DELAY commands, causing the exec to run slower.

Because DELAY PERCENT applies to all execs that run on a CA Verify for CICS port, an exec that calls other execs can control the speed of the called execs by issuing a DELAY PERCENT command.

At the completion of a DELAY command, updated information is returned in session variables that reflects any outbound message(s) received during a delay. Refer to CA Verify for CICS REXX Variables (see page 349) for a description of the information returned in session variables.

Conditions and Return Codes

At the completion of a DELAY command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes DELAY command return codes and REXX conditions that are raised by default when some CA Verify for CICS conditions occur.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
LOCKOUT	4		Timed out waiting for port lock
TIMEOUT	8		Timed out waiting for response
LOGOFF	16		Session Ended
ENDTRACE	20	HALT	Trace abended (ie., SD37)
CANCEL	24	HALT	CANCEL command processed
FAULT	64	HALT	Invalid command or system error

DELETEQ

The DELETEQ command is used to delete the entire data queue that is anchored to the port being accessed by the issuing exec.

Example:

```
" DELETEQ     "
```

Format:

```
  DELETEQ
```

Conditions and Return Codes

At the completion of a DELETEQ command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes DELETEQ command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
LOCKOUT	0		Timed out waiting for port lock
CANCEL	24	HALT	CANCEL command processed
FAULT	64	HALT	Invalid command or system error

ENDTRACE

The ENDTRACE command ends a session trace anchored to the port being accessed by an exec when the command is issued. A session trace is started by the TRACE command.

Example:

```
" ENDTRACE    "
```

Format:

```
   ENDTRACE
```

Conditions and Return Codes

At the completion of an ENDTRACE command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes ENDTRACE command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
LOCKOUT	0		Timed out waiting for port lock
CANCEL	24	HALT	CANCEL command processed
FAULT	64	HALT	Invalid command or system error

HANDLE

The HANDLE command activates or inactivates automatic handling of named session events when they occur during the processing of subsequently executed CA Verify for CICS session commands.

Examples:

```
" HANDLE turnaround off    "" HANDLE outbound off      "
```

Format:

```
HANDLE event option
```

event

One of the keywords, TURNAROUND or OUTBOUND, that names the session event for which handling is being specified.

Option

One of the keywords, ON or OFF, that specifies whether automatic handling of the named event is to be activated or inactivated.

The HANDLE command TURNAROUND keyword refers to the point in a session that occurs when a TYPE command has sent input to an online system and changes the session direction to outbound. By default or when HANDLE TURNAROUND ON is specified, the TYPE command automatically waits for and receives the online system response to a terminal input. When HANDLE TURNAROUND OFF is specified, subsequent TYPE commands do not wait for the online system response but rather complete immediately after the session direction is changed to outbound. HANDLE TURNAROUND OFF makes the TYPE command a send-only operation. After executing a TYPE command while TURNAROUND handling is off, an exec must eventually use the INVITE command to receive the online system response to the input sent by a TYPE command.

The HANDLE command OUTBOUND keyword refers to the point in a session at which an outbound message has been received from an online system that does not free the terminal keyboard, but rather leaves the session in the outbound state. By default or when HANDLE OUTBOUND ON is specified, when a CA Verify for CICS session command receives a message from an an online system that leaves the session in the outbound state, then CA Verify for CICS automatically waits for and receives the next message sent by the online system. When HANDLE OUTBOUND OFF is specified, TYPE and INVITE commands that receive messages that leave the session in the outbound state complete following the receipt of each message, whether or not the terminal keyboard is freed. After executing a TYPE or INVITE command that ends a message, is received, and leaves the session in an outbound state while OUTBOUND handling is off, an exec must eventually use the INVITE command to receive the next message sent by the online system.

When HANDLE commands are used to obtain control of outbound messages, and when TYPE and INVITE commands complete while the virtual terminal keyboard is not free, an exec must be aware of the state of a session and act accordingly. Specifically, a TYPE command is valid only while the keyboard is free. An INVITE command must eventually be issued to receive an online system message while the keyboard is locked. Execs that choose to handle outbound message flows may use CA Verify for CICS-defined variable fields PTEKB or PTESTATE to determine the state of a session and to decide whether TYPE commands are valid or INVITE commands are required. Refer to CA Verify for CICS REXX Variables for descriptions of PTEKB or PTESTATE and the possible values of the variable fields.

Conditions and Return Codes

At the completion of a HANDLE command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the HANDLE command. The following table describes HANDLE command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
CANCEL	24	HALT	CANCEL command processed
FAULT	64	HALT	Invalid command or system error

INVITE

The INVITE command receives any message sent by an online system within a specified time interval. It may be used to receive a message that is sent by an online system while the terminal keyboard is free, or to receive messages when the state of a session is outbound after using HANDLE OUTBOUND, HANDLE TURNAROUND, or WAIT time ON RESPONSE commands.

Examples:

```
" INVITE 1000 "" INVITE &invtime "
```

Format:

```
INVITE hhmmssth
```

hhmmssth

An optional time interval, expressed in hours (hh), minutes (mm), seconds (ss) and tenths and hundredths of seconds (th).

The INVITE command is used to receive a message at a virtual terminal that is anchored to the port that is being accessed by an exec. The LOGON command allocates and anchors a virtual terminal to an accessed port and uses the terminal to start a session.

A CA Verify for CICS LOGON command completes immediately after a session is started. The virtual terminal keyboard is always free at the beginning of a session, as is the keyboard on a real terminal. While it is true that a terminal keyboard is free before the first message is sent on a new session, this is seldom noticed by a person. After starting a session, most online systems send a message to the terminal. People are naturally patient and wait until they see the expected message.

REXX execs that use CA Verify for CICS host commands to start and use terminal sessions with an online system take the place of a terminal operator and must do the things that a person would do if a session is to flow smoothly. The INVITE command allows an exec to wait for an expected message, as a person would. At the start of a session, if an online system sends one or more initial messages, an exec must execute one or more INVITE commands to allow the online system to send the messages.

After receiving a terminal input, some online systems and applications send a message that frees the terminal keyboard immediately after receiving the terminal input. Then, at a later time, the online system sends another message that a terminal user sees as the response to the terminal input. Like a signon prompt sent at the start of a session, the unnoticed free keyboard situation does not cause a problem because the terminal operator is patient and waits for the expected message. A TYPE command ends when an online system message frees the terminal keyboard. If an online system sends a response that frees the terminal keyboard before sending the response message the user is expecting, then an exec must add patience by executing INVITE commands whenever it is necessary to wait for the delayed response.

Because people do not notice some free keyboard situations, it is sometimes difficult to determine when to code INVITE commands. One way to determine if an INVITE is needed is to examine a CA Verify for CICS test stream for the online system or application in question. A test stream may be created using option L, Log a Test Stream, on the CA Verify for CICS menu.

Browse the test stream, stopping at the record selection list. Enter the PROFILE command, select the WCC category, press PF3 to return to the record selection list. Paying attention to the OP column (operation), scroll forward through the record selection list. If you see two or more output operations in a row - without input operations between each output, an INVITE may be needed. Any WCC value greater than C1 unlocks the keyboard. If the keyboard is unlocked before the last output in the sequence, an INVITE is recommended after each output after the keyboard was unlocked.

At the completion of an INVITE command, updated information is returned in session variables, as documented in CA Verify for CICS REXX Variables.

Conditions and Return Codes

At the completion of an INVITE command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes INVITE command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
LOCKOUT	4		Timed out waiting for port lock
TIMEOUT	8		Timed out waiting for response
LOGOFF	16		Session ended
ENDTRACE	20	HALT	Trace abended (ie., SD37)
CANCEL	24	HALT	CANCEL command processed
FAULT	64	HALT	Invalid command or system error

LOCK

The LOCK command locks the port that is being accessed by an exec and all resources anchored to the port.

The ACCESS command may also be used to lock a port when it is accessed.

While a port is locked, resources anchored to the port may be used only by execs that run on same port as the exec that issued the LOCK command. If an exec running on any other port issues a CA Verify for CICS command that uses resources anchored to the locked port, the command is delayed until the port is unlocked.

The UNLOCK command is used to unlock a locked port.

Example:

```
" LOCK  "
```

Format:

```
LOCK
```

Note: Refer to Sharing Resources with Attached Execs (see page 288) for more information about locks and uses of locks.

Conditions and Return Codes

At the completion of an LOCK command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes LOCK command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
LOCKOUT	0		Timed out waiting for port lock
CANCEL	24	HALT	CANCEL command processed
FAULT	64	HALT	Invalid command or system error

LOGOFF

The LOGOFF command ends a session that was started using the LOGON command.

Example:

```
" LOGOFF "
```

Format:

```
LOGOFF
```

The LOGOFF command ends the session anchored to the port being accessed by an exec when the command is executed.

When the LOGOFF command is used to end a session, the online system sees the session termination in exactly the same way that it sees a session termination that occurs when a user presses the SYSREQ key on a real terminal keyboard, then enters the command LOGOFF, to request that VTAM terminate the active application session.

The preferred way to end a session is to issue a TYPE command that enters a particular online system's logoff command or transaction. Entering a logoff transaction may be required for some online systems to give the online system an opportunity to clean up resources that were allocated when the user signed on.

At the completion of a LOGOFF command, all session variables described in CA Verify for CICS REXX Variables are dropped.

Conditions and Return Codes

At the completion of a LOGOFF command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes LOGOFF command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
LOCKOUT	4		Timed out waiting for port lock
ENDTRACE	20	HALT	Trace abended (ie., SD37)

Conditions	RC	Default Signal	Reason for Condition
CANCEL	24	HALT	CANCEL command processed
FAULT	64	HALT	Invalid command or system error

LOGON

The LOGON command allocates a CA Verify for CICS virtual terminal, anchors the terminal to the port being accessed by the exec when the LOGON command is executed, and uses the terminal to start a session with a named online system.

Examples:

```
" LOGON cics1 model mod2 "" LOGON &system data &uid  "
```

Format:

```
LOGON system-name (MODEL terminal-model | MOD2 | MOD3 | MOD4 | MOD5 ) (
default-screen-size24X80 | 32X80 | 43X80 | 27X132 ) ( alternate-screen-size24X80 |
32X80 | 43X80 | 27X132 )| ( terminal-type

  BASICDS | EXTDS )

 (FROM terminal-name )

 (DATA logon-data )

 ( buffer-format

BASICDATA | EXTDSDATA | PANELDATA | IMAGEDATA |  ATTRIBUTES attr-values )( USERMOD
ispf-usermod )
```

system-name

The 1 through 8 character name of an online system with which a session is to be started. The system name specified may be an actual name, a literal value or a REXX variable, coded with a leading ampersand, that contains the name of an online system.

MODEL

An optional keyword and indicates that the value that follows is a symbolic terminal model name or the name of a mode table entry.

terminal-model

The logon mode table entry to be used to start a session. It may be one of the keywords MOD2, MOD3, MOD4 or MOD5, to refer to a mode table entry name, or it may be the actual name of a logon mode table entry. terminal-model may be expressed as an actual value, or may be a variable field that contains a valid terminal model value or null value.

default-screen-size

(Optional) Indicates the default virtual terminal screen size to be used for a session. default-screen-size may be one of the keywords 24X80, 32X80, 43X80, or 27X132, expressed as an actual value, or as a variable field that contains a screen size keyword or a null value.

alternate-screen-size

(Optional) Indicates the alternate virtual terminal screen size to be used for a session. alternate-screen-size may be one of the keywords 24X80, 32X80, 43X80, or 27X132, expressed as an actual value, or as a variable field that contains a screen size keyword or a null value.

terminal-type

(Optional) Specifies the type of terminal to be used for a session. terminal-type may be one of the keywords BASICDS or EXTDS, expressed as an actual value, or as a variable field that contains a terminal type keyword or null value.

FROM

An optional keyword that indicates that the data item that follows is a virtual terminal name.

terminal-name

The name of the virtual terminal to be used for the session. terminal-name may be the actual name of a VTAM APPL that has been defined at an installation for use as a CA Verify for CICS virtual terminal or may be a variable field that contains a virtual terminal name or a null value. If the FROM keyword is not specified or if terminal-name is a variable field that contains a null value, then a virtual terminal will be assigned automatically.

DATA

An optional keyword that indicates that the data item that follows is logon data.

logon-data

A data item identified by the DATA keyword that is to be sent to an online system with a session request. logon-data may be a variable field that contains logon data and may be up to 255 characters in length, or may be one or more consecutive literal values whose accumulated length(s) do not exceed 255 characters. When the DATA keyword is not specified or if logon-data is a variable field that contains no data, then data is not included in the session request sent to an online system.

buffer-format

An optional keyword that specifies the format of the virtual terminal display buffer data returned in REXX variables PTEBUFF and PTEEAB at the completion of the LOGON command and subsequent session commands. Refer to 3270 Attribute Values for the values and meanings of basic, panel and extds attributes and extended attribute data.

- By default, basic field attribute values are returned in PTEBUFF and extended attribute data is returned in PTEEAB only if an EXTDS session is active.
- BASICDATA returns basic attributes in PTEBUFF and does not return PTEEAB, even if an EXTDS session is active.
- EXTDSDATA returns extds attributes in PTEBUFF and returns extended attribute data in PTEEAB, even if a BASICDS session is active.
- PANELDATA returns panel attributes in PTEBUFF, blanks out protected non-display data and translates nulls in protected display fields to blanks. PTEEAB is not returned.
- IMAGEDATA translates all non-display data, attributes and and nulls to blanks in the returned PTEBUFF, and does not return PTEEAB. IMAGEDATA produces a legible PTEBUFF that may be printed or used in application documentation.
- ATTRIBUTES may be used to specify the 32 1-byte hexadecimal values to be returned as the 32 possible basic 3270 field attributes in PTEBUFF. The value following ATTRIBUTES, attribute-values, is a 64-character literal value enclosed in single or double quotes. Each character is a hexadecimal character, 0-F,and each successive pair of characters is a hexadecimal value to be used to represent the next successive basic field attribute. When ATTRIBUTES is specified, PTEEAB is returned if an EXTDS session is active.

USERMOD

An optional keyword that indicates that the value that follows is an ISPF usermod value.

ISPF-usermod

Immediately follows the USERMOD keyword and specifies a one-byte value, expressed as a single character or as a two-character hexadecimal value enclosed in single or double quotes. The ISPF-usermod value tells CA Verify for CICS the USERMOD value specified by an ISPF panel that is used to display an image of a virtual terminal screen. ISPF-usermod defaults to hexadecimal 3F.

The system name specified may be the actual VTAM APPL name by which an online system is known to VTAM or may be a familiar system id defined in CA Verify for CICS system options, that refers to an actual VTAM APPL.

If a null terminal model is specified in a CA Verify for CICS LOGON command, or if the MODEL keyword is not specified, then a default mode table entry name specified during CA Verify for CICS installation is used. The following table lists terminal model keywords and their associated default and alternate virtual terminal screen sizes.

Model Keyword	Default Size	Alternate size
MOD2	24 X 80	24 X 80
MOD3	24 X 80	32 X 80
MOD4	24 X 80	43 x 80
MOD5	24 X 80	27 x 132

The first screen size keyword encountered in a LOGON command specifies the default screen size. If a default screen size is specified, it is used only when a session is started using a mode table entry that does not contain screen sizes, and sizes are not forced by an online system.

The second screen size keyword encountered in a LOGON command specifies the alternate screen size. If an alternate screen size is specified, it is used only when a session is started using a mode table entry that does not contain screen sizes, and sizes are not forced by an online system.

If screen sizes are not specified and are not defined in the mode table used to start a session and are not forced by an online system, then default sizes specified at an installation are used.

A terminal may be either a basic or extended 3270 data stream terminal. A basic data stream terminal has a single screen buffer that is used to store screen data and coded basic field attributes. An extended data stream terminal has a screen buffer that is used to store screen data and actual basic field attributes and has an extended attribute buffer that is used to store extended data stream color and highlighting attributes and other information.

If a terminal type is not specified, a default type specified at an installation is used. If an online system sends a 3270 Read Partition Query structured field to a terminal to determine device capabilities, the terminal type or screen sizes specified influence the contents of the reply.

After using the LOGON command to start a session, the CA Verify for CICS TYPE, INVITE, and DELAY commands are used to key data into the virtual terminal used for a session, to send input to the online system, to receive online system responses to input, and to receive messages initiated by an online system.

ISPF Session Panels explains how to use the LOGON command PANELDATA and USERMOD options to display an image of a virtual terminal screen using an ISPF panel, and how to key data entered by the user into a virtual terminal using the TYPE command USERMOD option.

At the completion of each CA Verify for CICS session command, session information is returned to the issuing exec in REXX variables. Refer to CA Verify for CICS REXX Variables for a description of each session variable.

A session is ended by an online system when a logoff transaction is entered using a TYPE command, or may be ended using a LOGOFF command. Sessions are terminated automatically if all execs in an address space end and leave sessions active.

Conditions and Return Codes

At the completion of a LOGON command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes LOGON command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
LOCKOUT	4		Timed out waiting for port lock
LOGOFF	16		Session ended
ENDTRACE	20	HALT	Trace abended (ie., SD37)
CANCEL	24	HALT	CANCEL command processed
FAULT	64	HALT	Invalid command or system error

MONITOR

The MONITOR command instructs CA Verify for CICS to monitor or to stop monitoring specified events and specifies what to do when monitored events are detected. Detected events may be displayed at a terminal, written as REXX SAY messages, or may passed to a user-written exec that may process the event as desired. MONITOR command instructions apply to subsequent CA Verify for CICS commands issued by any exec that runs in the REXX Language Processor Environment in which the MONITOR command is issued, including execs that call or that are called by an exec that issues a MONITOR command. A MONITOR command may instruct CA Verify for CICS to monitor or to stop monitoring:

- CA Verify for CICS commands issued by REXX execs
- online system responses or other messages received by CA Verify for CICS session commands
- REXX variable values assigned or used by CA Verify for CICS commands
- REXX conditions raised by CA Verify for CICS commands to indicate CA Verify for CICS conditions to an exec
- CA Verify for CICS conditions returned to an exec as a return code

Examples:

```
" MONITOR SIGNALS "

" MONITOR SIGNALS CONDITIONS COMMANDS RESPONSES STORE DROP FETCH "

" MONITOR &REXXSYM &PTEMESS "

" MONITOR SIGNALS CONDITIONS COMMANDS RESPONSES VIAEXEC MONEXEC "

" MONITOR OFF "
```

Format:

```
MONITOR ( events )( &generic_symbols )( OFF )( VIASAY | VIATPUT | VIAEXEC monitor_exec
)( NOTE &note | RECALL &note )
```

events

An optional list of keywords that name events to be monitored. When OFF is also specified monitoring is stopped for the specified events. Event keywords are:

- SIGNALS monitors REXX HALT, ERROR or FAILURE conditions raised by CA Verify for CICS commands to indicate CA Verify for CICS conditions to an exec.
- CONDITIONS monitors return codes returned to execs by CA Verify for CICS commands to indicate CA Verify for CICS LOCKOUT, TIMEOUT, ENDQUEUE, LOGOFF, ENDTRACE, CANCEL or FAULT conditions.
- COMMANDS monitors the start of subsequent commands.
- RESPONSES monitors online system responses and other messages sent by an online system to a virtual terminal, as well as session initiation and termination events, when they are received by CA Verify for CICS session commands. When RESPONSES are received, information is not displayed; instead a specified or default monitor_exec is called to process the information. The distributed default monitor_exec, VTEMONX, shown in The Distributed Monitor Exec (see page 328) displays commands and virtual terminal screen images using ISPF display services or the REXX SAY instruction, depending on the REXX environment.
- STORE monitors changes to variables by CA Verify for CICS commands or monitors changes to variable values whose names begin with specified &generic_symbols.
- FETCH monitors uses of variables by CA Verify for CICS commands or monitors uses of variables whose names begin with specified &generic_symbols.
- DROP monitors variables dropped by CA Verify for CICS commands or monitors dropped variables whose names begin with specified &generic_symbols.

&generic_symbols

Monitors operations performed by CA Verify for CICS commands on REXX variables whose names begin with any specified generic_symbol, or stops monitoring when OFF is also specified. For example, MONITOR &PTE &Q_VALUE, monitors CA Verify for CICS operations on variables whose names begin with PTEor Q_VALUE. When STORE, FETCH and/or DROP are also specified, only the specified operations are detected, otherwise all operations on variables whose names begin with specified &generic_symbols are detected.

OFF

Stops monitoring for specified events and/or &generic_symbols. If OFF is the only MONITORcommand option specified, then all monitoring is stopped.

VIASAY

Causes messages to be sent to the REXX SAY destination when monitored events other than RESPONSES are detected.

VIATPUT

Causes messages to be sent to the TSO terminal using TSO TPUT, when monitored events other than RESPONSES are detected. In a non-TSO address space, VIATPUT sends messages to the REXX SAY destination.

VIAEXEC monitor_exec

Specifies that a named exec, monitor_exec, is to be called when a monitored event is detected. The Monitor Exec Interface (see page 327) documents arguments passed to a monitor_exec when it is called to process events and provides other information about monitor execs.

NOTE &note

Saves the value of REXX variable note for subsequent recall by any exec that runs on the same port as the exec that saved the &note.

RECALL &note

Returns a monitor note in REXX variable note that was saved by a MONITOR NOTE command issued by any exec running on the same port as the exec that recalls the &note.

By default, CA Verify for CICS commands are not monitored. While return codes, REXX conditions and information returned in REXX variables informs an exec about errors or other CA Verify for CICS conditions encountered by CA Verify for CICS commands, no information about CA Verify for CICS conditions is automatically displayed or otherwise made available to the user of an exec.

- A MONITOR command that specifies only SIGNALS may be used to display or SAY CA Verify for CICS error messages automatically.

- A MONITOR command that specifies RESPONSES may be added to a converted or user written exec that uses CA Verify for CICS session commands to automatically display the session while the exec is executing.
- MONITOR commands that specify other events and/or variables may be used to perform more extensive exec debugging.

Because MONITOR commands apply to all execs that run in a REXX environment, general purpose execs that activate monitoring and call other execs may be used to debug execs without modification to the exec being debugged. In addition to debugging applications, the MONITOR command, monitor execs, other CA Verify for CICS commands, and ISPF commands may be used to monitor multi-session stress tests. More information about and examples of MONITOR command applications may be found in Monitoring Execs and Sessions (see page 293).

Conditions and Return Codes

At the completion of a MONITOR command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes MONITOR command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
CANCEL	24	HALT	CANCEL command processed
FAULT	64	HALT	Invalid command or system error

The Monitor Exec Interface

A monitor exec specified using the MONITOR command VIAEXEC option is called to process all monitored events when they are detected. A default monitor exec is called to process online system responses when RESPONSES are monitored, but a monitor exec has not been specified using the MONITOR command VIAEXEC option. When a monitor exec is called to process an event, information about the event is passed to the monitor exec as CALL arguments. A monitor exec may assign the argument values to REXX variables using a REXX ARG instruction that has commas coded between specified variable names. The CALL arguments passed to a monitor exec are:

- The name of the REXX Environment in which the monitored event was detected, as specified by REXX parameters when the environment was initialized, possibly z/OS, TSO/E or ISPF.
- The name of the REXX exec that issued the CA Verify for CICS command that was being processed when the monitored event was detected.
- The text of the CA Verify for CICS command being executed when the monitored event was detected.
- One of the keywords, COMMAND, RESPONSE, STORE, FETCH, DROP or CC, that identifies the event to be processed by the monitor exec.
- When argument 4 is STORE, FETCH or DROP, then argument 5 is the name of the REXX variable whose value was stored, fetched or dropped.
- When argument 4 is CC, then argument 5 is one of REXX conditions HALT, ERROR or FAILURE, if a REXX condition is being raised by a CA Verify for CICS command, or is RC if a return code is being returned to an exec to indicate a CA Verify for CICS condition.
- When argument 4 is STORE or FETCH, then argument 6 is the value of the REXX variable that was stored or fetched.
- When argument 4 is CC, then argument 6 is one of the CA Verify for CICS condition keywords, LOCKOUT, TIMEOUT, ENDQUEUE, LOGOFF, ENDTRACE, CANCEL or FAULT, and indicates why a REXX condition is being raised or a return code is being returned to an exec.
- When argument 4 is CC, argument 7 is a CA Verify for CICS message that further documents why a REXX condition is being raised or a return code is being returned to an exec.

Prior to calling a monitor exec, active monitoring is turned off, and monitoring is restored when a monitor exec returns. A monitor exec may issue an ACCESS LASTPORT command to determine the port being accessed by the exec when the monitored event was detected, may issue a QUERY SESSION command to fetch a copy of all CA Verify for CICS defined session variables, or may issue any other CA Verify for CICS command including the MONITOR command. Monitor execs may also ADDRESS other command processors and use other commands that are valid in the REXX Language Processor Environment.

MONITOR NOTE and MONITOR RECALL functions are intended for use by monitor execs and other execs that control monitor functions and monitor execs. An exec that issues a MONITOR command that specifies a user-written monitor exec may pass information to the monitor exec using a monitor note. A monitor exec may recall information noted by another exec or may recall, update and note session statistics or other information needed when the monitor exec is called to process subsequent events. While there are no restrictions regarding the use of the MONITOR NOTE or RECALL functions, it is recommended that the functions be used only by monitor execs or execs that communicate or cooperate with monitor execs, to avoid conflicting use of notes if monitor execs are used to monitor other execs.

The Distributed Monitor Exec

If a MONITOR command specifies RESPONSES, but does not specify VIAEXEC, a default monitor exec specified in CA Verify for CICS system options is called when subsequent online system responses or other outbound session events are processed by CA Verify for CICS session commands.

A distributed monitor exec, VTEMONX, is designed to be used as a default monitor exec. When called to process detected RESPONSE events, VTEMONX displays CA Verify for CICS session commands and virtual terminal screen images using ISPF display and pop-up services, when called in an ISPF REXX Language Processor Environment, or using the REXX SAY instruction, when called in a TSO or other REXX Language Processor Environment.

VTEMONX is distributed in the CA Verify for CICS CATJEXEC library. ISPF panels used by VTEMONX, VTEPXMDT and VTEPXMW2, are distributed in the CA Verify for CICS CATJPENU library.

VTEMONX should not be specified as a MONITOR command VIAEXEC. If called to process an event other than a RESPONSE event, VTEMONX ignores the event and returns. Monitored events other than RESPONSE events are displayed automatically if a VIAEXEC is not specified. However, a user-written monitor exec that processes other events may call VTEMONX to process RESPONSE events provided that it passes all arguments passed to it by CA Verify for CICS.

POST

The POST command replaces the entire queue anchored to the port being accessed by an exec, with a single queue entry that contains specified argument values. If a PULL command issued by an exec running on another port is waiting for data to be stored in the queue, then the oldest waiting PULL command is resumed when the queue entry is posted.

Examples:

```
" POST 'saymsg' &PTEMESS "
" POST &item1 &item2 &item3 "
```

Format:

```
 POST (arg1 . . . arg20)
```

arg1 through arg20

(Optional) Argument which specifies up to 20 values, either literal values contained in single or double quotes, or REXX variables, coded with a leading ampersand, that contain argument values to be stored in the posted queue entry. A null queue entry is posted if no arguments are specified.

If the POST command is used to post updated information and QUERY QUEUE commands are used to examine the information, a queue becomes a bulletin board that may be used to provide current information to any number of other execs. QUEUE and PULL commands may be used to process queue entries in the order that they were queued. PUSH and PULL commands may be used to process the most recently queued entry first.

Conditions and Return Codes

At the completion of a POST command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
LOCKOUT	4		Times out waiting for port lock
CANCEL	24	HALT	CANCEL command processed
FAULT	34	HALT	Invalid command or system error

PULL

The PULL command assigns argument values fetched from the top entry of the queue anchored to the port being accessed by an exec when the command is issued, to variables specified by the PULL command.

Example:

```
" PULL &var1 &var2 &var3 "
```

Format:

```
PULL (&variable1 . . . &variable20)
```

&variable1 through &variable20

The names of up to 20 variables, coded with a leading ampersand, to which pulled argument values may be assigned.

Argument values stored in the pulled queue entry by a QUEUE, PUSH, or POST command are returned in the corresponding variables specified by a PULL command.

If the number of variables specified exceeds the number of argument values in the pulled queue entry, then a zero-length value is assigned to excess variables. If the number of argument values in a pulled queue entry exceeds the number of variables specified by the PULL command, then excess argument values are discarded.

A CA Verify for CICS WAIT command that specifies ON PULL may be issued to specify a time interval that subsequent PULL commands are to wait for data to be queued in the event that an accessed queue is empty when a PULL command is issued. A PULL completes normally if queued data is immediately available or if data is queued within a specified time interval. If an accessed queue is empty and a WAIT interval was not specified or a specified interval expires before data is queued, the PULL command completes with return code 12 to indicate the CA Verify for CICS ENDQUEUE condition.

Conditions and Return Codes

At the completion of a PULL command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes PULL command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully

Conditions	RC	Default Signal	Reason for Condition
LOCKOUT	4		Times out waiting for port lock
ENDQUEUE	12		Data queue was empty
CANCEL	24	HALT	CANCEL command processed
FAULT	34	HALT	Invalid command or system error

PUSH

The PUSH command adds an entry to the top of the queue anchored to the port being accessed by an exec when the command is issued, that contains specified argument values. If a PULL command issued by an exec running on another port, is waiting for data to be stored in the queue, then the oldest waiting PULL command is resumed when an entry is queued by a PUSH command.

Examples:

```
" PUSH 'INQUIRE' &ACCOUNT  "

" PUSH &item1 &item2 &item3 "
```

Format:

```
PUSH (arg1 . . . arg20)
```

arg1 through arg20

(Optional) Argument which specifies up to 20 values, either literal values contained in single or double quotes, or REXX variables, coded with a leading ampersand, that contain argument values to be stored in the queue entry. A null entry is queued if no arguments are specified.

When the PUSH command is used to queue data and the PULL command is used to process it, the most recently queued entries are processed first. QUEUE and PULL commands may be used to process queue entries in the order that they were queued. A queue may also be used as a bulletin board to post current information that may be examined to any number of other execs using the POST and QUERY QUEUE commands.

Conditions and Return Codes

At the completion of a PUSH command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes PUSH command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
LOCKOUT	4		Times out waiting for port lock
CANCEL	24	HALT	CANCEL command processed
FAULT	34	HALT	Invalid command or system error

QUERY

The QUERY command returns requested information about CA Verify for CICS resources in REXX variables. The QUERY command may be used to:

- Obtain information about the CA Verify for CICS session, session trace, data queue or exec task on the port being accessed by the interrogating exec.
- Obtain a list of ports that have active CA Verify for CICS sessions, active session traces, queued data, or active REXX exec tasks.

Examples:

```
" QUERY session          "

" QUERY sessions         "

" QUERY queue &var1 &var2 &var3 "
```

Format:

```
QUERY
    ( QUEUE (&variable1 . . . &variable20) | QUEUES | SESSION ( LOGONDATA |
   BASICDATA | EXTDSDATA |
   PANELDATA | IMAGEDATA ) | SESSIONS | TASK | TASKS |
  TRACE | TRACES )
```

QUEUE, QUEUES, SESSION, SESSIONS, TASK, TASKS, TRACE and TRACES

Keywords that identify the CA Verify for CICS resource or resources about which information is to be returned.

&variable1 . . . &variable20

Can be specified when a QUEUE is queried, they are the names of up to 20 REXX variables, coded with a leading ampersand, to which the queried argument values are to be assigned.

LOGONDATA, BASICDATA, EXTDSDATA, PANELDATA and IMAGEDATA

Keywords that may be specified when a SESSION is queried to specify the format of the virtual terminal display buffers returned in variables PTEBUFF and PTEEAB.

When QUEUES, SESSIONS, TASKS or TRACES is specified, NOQUEUES, NOSESSIONS, NOTASKS or NOTRACES is returned in variable, PTEINFO, if the queried resource does not exist on any CA Verify for CICS port. Otherwise QUEUES, SESSIONS, TASKS, or TRACES is returned in PTEINFO and a list of the ports on which the queried resource exists is returned in PTEPORTS. For example:

- If only the queues on ports 1, 2, and 6 contain data, the command, query queues, returns QUEUES in PTEINFO and returns 1 2 6 in PTEPORTS.
- If all queues are empty, NOQUEUES is returned in PTEINFO.
- Query tasks, never returns NOTASKS in PTEINFO, because at least the exec that issued the QUERY command is active on a port.

When QUEUE, SESSION, TASK or TRACE is specified, NOQUEUE, NOSESSION, NOTASK or NOTRACE is returned in PTEINFO, if the queried resource is not active on the port being accessed by the exec when the QUERY command is issued. Otherwise QUEUE, SESSION, TASK or TRACE is returned in PTEINFO.

When a QUERY QUEUE command specifies one or more variables, &variable1 through &variable20, and the accessed queue contains data, then values obtained from the top entry in the queue are returned in corresponding variables specified by the QUERY command. If the number of variables specified exceeds the number of values in the top queue entry, a zero-length value is returned in excess variables. The QUERY QUEUE command works like the PULL command, except that the QUERY command does not remove data from a queue.

A QUERY SESSION command returns the same information in session variables that is returned by all CA Verify for CICS session commands, and drops session variables if a session is not active on the accessed port. However, a QUERY SESSION command may specify one of the keywords, LOGONDATA, BASICDATA, EXTDSDATA, PANELDATA or IMAGEDATA, to override the format of the virtual terminal display buffers returned by the QUERY command in PTEBUFF and PTEEAB. By default or when LOGONDATA is specified, buffer data is returned using the format specified by the LOGON command that started the queried session. Refer to LOGON for a description of the format of the buffer data returned when other optional format keywords are specified. Refer to CA Verify for CICS REXX Variables for a description of all session variables.

Conditions and Return Codes

At the completion of a QUERY command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes QUERY command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
LOCKOUT	4		Times out waiting for port lock
CANCEL	24	HALT	CANCEL command processed
FAULT	34	HALT	Invalid command or system error

QUEUE

The QUEUE command adds an entry to the bottom of the queue anchored to the port being accessed by an exec when the command is issued, that contains specified argument values. If a PULL command issued by an exec running on another port is waiting for data to be stored in the queue, then the oldest waiting PULL command is resumed when the queue entry is added.

Examples:

```
" QUEUE 'INQUIRE' &ACCOUNT  "

" QUEUE &item1 &item2 &item3 "
```

Format:

```
QUEUE (arg1 . . . arg20)
```

arg1 through arg20

(Optional) Arguments which specify up to 20 values, either literal values contained in single or double quotes or REXX variables, coded with a leading ampersand, that contain argument values to be stored in the queue entry. A null entry is queued if no arguments are specified.

When the QUEUE command is used to queue data and the PULL command is used to process it, queue entries are processed in the order they were queued. PUSH and PULL commands may be used to to process the most recently queued entry first. A queue may also be used as a bulletin board to post current information that may be examined to any number of other execs using the POST and QUERY QUEUE commands.

Conditions and Return Codes

At the completion of a QUEUE command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes QUEUE command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
LOCKOUT	4		Times out waiting for port lock
CANCEL	24	HALT	CANCEL command processed
FAULT	34	HALT	Invalid command or system error

SIGNAL

The SIGNAL command instructs subsequent CA Verify for CICS commands to signal the occurrence of a specified CA Verify for CICS condition by raising a REXX ERROR, FAILURE or HALT condition, or to raise no REXX condition if a specified CA Verify for CICS condition is encountered.

Examples:

```
"          SIGNAL OFF ON ENDTRACE  "

"          SIGNAL ERROR ON LOGOFF  "
```

Format:

SIGNAL REXX-cond ON Verify-cond

REXX-cond is one of the keywords, ERROR, FAILURE, HALT or OFF.

- ERROR, FAILURE or HALT specify the REXX condition to be raised if the specified Verify-cond occurs, and are also the REXX condition keywords used in a REXX SIGNAL or CALL instruction to trap the CA Verify for CICS-cond.
- OFF specifies that no REXX condition is to be raised if the specified Verify-cond occurs.

ON is required and identifies the Verify-cond that follows.

Verify-cond is one of the CA Verify for CICS condition keywords, LOCKOUT, TIMEOUT, ENDQUEUE, LOGOFF, ENDTRACE, CANCEL or FAULT.

All unusual events and errors that can be detected by CA Verify for CICS commands are categorized into named groups called Verify conditions. Each Verify condition is associated with a unique return code that is returned in the REXX RC variable when a Verify condition is detected. The following table is a summary of all possible Verify condition and return codes and default REXX conditions raised for some CA Verify for CICS conditions.

The CA Verify for CICS SIGNAL command is designed for use with REXX SIGNAL and CALL instructions to trap CA Verify for CICS conditions using REXX condition traps, or may be used to avoid REXX conditions that are raised by default for some CA Verify for CICS conditions.

For example, instead of checking the return code following each CA Verify for CICS session command to detect and process an unexpected session failure, a CA Verify for CICS SIGNAL command and a REXX SIGNAL instruction may be used to detect and process an unexpected session failure.

If a session is traced, but a trace failure is not a big enough problem to interrupt an exec, the SIGNAL command may be used to override the REXX HALT condition that is raised by default when a CA Verify for CICS ENDTRACE condition occurs.

```
"  SIGNAL OFF ON ENDTRACE
"
"
  trace
  to tracedd
"
"
  SIGNAL ERROR ON LOGOFF
"
  signal ON ERROR name ended
"
  logon
  cics
"
   say 'session ended:' PTEMESS
```

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
LOCKOUT	4		Timed out waiting for port lock
TIMEOUT	8		Timed out waiting for response

Conditions	RC	Default Signal	Reason for Condition
ENDQUEUE	12		Data queue was empty
LOGOFF	16		Session ended
ENDTRACE	20	HALT	Trace abended
CANCEL	24	HALT	CANCEL command processed
FAULT	34	HALT	Invalid command or system error

Conditions and Return Codes

At the completion of a SIGNAL command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table defines the CA Verify for CICS conditions, return codes and default REXX conditions that may be raised by the SIGNAL command.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
CANCEL	24	HALT	CANCEL command processed
FAULT	34	HALT	Invalid command or system error

TRACE

The TRACE command starts a CA Verify for CICS session trace on the port being accessed by an exec when the command is issued. While a session trace is active, CA Verify for CICS writes the current image of a virtual terminal screen image to a trace data set whenever a TYPE command sends data to an online system and when an online system sends a response or other message to the virtual terminal being used for a session.

Examples:

```
" TRACE to tracedd image freekb  "

" TRACE 'change caption to this' "

" TRACE to &ddname format pack "
```

Format:

```
TRACE TO ddname
( FORMAT | IMAGE | UPPERCASE )
( DELAYS | DELAYSOFF )
( LOGONS | LOGONSOFF )
( CHAINS | FREEKB )
( PACK   | NOPACK )|
( caption )
```

TO ddname

Specifies the DD name to be opened to capture screen images when CA Verify for CICS session commands are used to send and receive data using the session anchored to the same port as the session trace.

FORMAT

(Optional) Keyword which specifies that traced screen images are to contain all screen data.

IMAGE

(Optional) Keyword which specifies that traced screen images are to be edited, to convert attributes and other non-display data to blanks and graphic escape characters to dashes, producing screen images similar to screen images visible to users of the system.

UPPERCASE

(Optional) Keyword which specifies that traced screen images are to be edited and converted to all upper case characters.

DELAYS

Specifies that user think time is to be captured as DELAY statements in the trace data set. DELAYSOFF specifies that no DELAY statements are to be created.

LOGONS

Specifies that LOGON statements are to be captured when a session is started. LOGONSOFF specifies that no LOGON statements are to be created.

CHAINS

Specifies that all outbound screen images are to be traced. FREEKB specifies that outbound screens images are to be captured only when an outbound message frees the terminal keyboard.

PACK

Specifies that traced session data is to be written using the ISPF packed data format. NOPACK specifies that traced screen images are to be written as unpacked, fixed-length records.

caption

a literal value that is to be used as the caption of subsequently traced screen images, and may be specified when a trace is started or at any time while a trace is active. A caption must be enclosed in either single or double quotes.

Default values for all optional parameters are controlled by CA Verify for CICS system options.

The TRACE command starts or modifies the trace on the port that is being accessed TRACE command is issued.

A trace may be started or ended before, during and after a session is started or ended to capture any portion of a session.

A trace is ended using the ENDTRACE command or is ended automatically when all execs in an address space have ended.

Conditions and Return Codes

At the completion of a TRACE command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes TRACE command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
LOCKOUT	4		Times out waiting for port lock
ENDTRACE	20	HALT	Trace abended (ie., SD37)
CANCEL	24	HALT	CANCEL command processed

Conditions	RC	Default Signal	Reason for Condition
FAULT	34	HALT	Invalid command or system error

TYPE

The TYPE command operates a virtual terminal by pressing keys on a virtual terminal keyboard that perform the same functions as the keys on a real terminal keyboard. For example, a tab key moves the cursor to the next modifiable field. When a data key is pressed, each character of the data is keyed in at the current cursor position, the modified data tag is turned on in the attribute that controls the field, and the cursor advances one position or skips to the next field if the position is a skip attribute.

A TYPE command uses the keyboard of the virtual terminal that is anchored to the CA Verify for CICS port being accessed by an exec when a TYPE command is issued, to position the cursor, to type literal, variable and other data values, to erase fields, to send data to an online system, and to perform other keyboard operations specified by the command. The keyboard operations specified by a TYPE command are performed in the order in which they are coded.

Examples:

```
" TYPE home erase 'cemt set'  "," ' da(&dd) open ena' enter "

" TYPE clear  "

" TYPE <2 15> 'inq'  "

" TYPE <4 15> 'smith' enter "

" TYPE <&row &col> &data1  "" TYPE <4 15> &data2 enter "

" TYPE home tab &account "," tab &order pf2 "

" TYPE usermod <&cursor> "pfkey
```

Format:

```
TYPE ( <row column> )
     ( <position> )
     ( HOME TAB BACKTAB NEWLINE )
     ( 'literal-data' )
     ( "literal-data" )
     ( &variable-data )
     ( ERASE )
     ( SELECT | DESELECT )
     ( TOKEN token )
     ( CODE code )
     ( USERMOD )
     ( ENTER | PF1-24 | PA1-3 |
     PEN | CLEAR | ATTN |AID &aid-variable )
     TYPE
```

<row column>

Moves the cursor to the specified virtual terminal screen row and column. A row or column may be a row or column number or the name of a REXX variable, coded with a leading ampersand, whose value is a row or column number.

<position>

Moves the cursor to the specified virtual terminal screen position, where position = (row - 1) * PTECOLS + column. position may be a number or the name of a REXX variable, coded with a leading ampersand, whose value is a screen position.

HOME, TAB, BACKTAB and NEWLINE

Move the cursor to the first modifiable screen field, to the field that follows or precedes the current cursor location, or to the first modifiable screen position that follows the current cursor row.

literal-data,

Types the indicated data, which must be enclosed in single or double quotes.

&variable-data

Types the data contained in a named REXX variable, coded with a leading ampersand.

ERASE

Erases the remainder of the modifiable screen field, beginning at the current cursor position.

SELECT and DESELECT

Keywords select or deselect the pen detectable selection or attention field at the current cursor location. The operations simulate operations performed by a 3270 light pen or cursor select key, with the following exceptions:

- SELECT selects a pen selectable field even if the field is already selected,
- DESELECT deselects a pen selectable field even if the field is already deselected,
- If a pen attention field with a null designator is selected, a PEN keyword must be coded as the last TYPE command keystroke to send the pen aid and data.
- If a pen attention field with an ampersand designator (a simulated ENTER key) is selected, an ENTER keyword must be coded as the last TYPE command keystroke to send the data.

TOKEN

Identifies a CA Verify for CICS password token, which follows the TOKEN keyword. token may be a literal value or a REXX variable coded with a leading ampersand, that contains a token (for example, &token). A token represents a password that is encrypted as described in Token Security. A token:

- is used to secure password data while Token Security is active;
- is converted to another form if Token Security is not active;
- may not be typed into a display field;
- is hidden in memory when it is typed;
- is not returned in PTEBUFF;
- can be decrypted only when used by its owner;
- is decrypted only when it is sent to an online system; and
- replaces password data when a session is traced.

CODE

Identifies a CA Verify for CICS password code, which follows the CODE keyword. code may be a literal value or a REXX variable, coded with a leading ampersand, that contains a code (for example, &code). A code is a password or other data that is encoded as described in Code Security. A code:

- is used to secure encodable passwords while Code Security is active;
- is converted to another form if Code Security is not active;
- may not be typed into a display field;
- can be decoded correctly only when used by its owner;
- is decoded when it is typed into a non-display field; and
- replaces encodable passwords when a session is traced;

USERMOD

Fetches data from fields of the virtual terminal display buffer in REXX variable, PTEBUFF, whose attribute values match a USERMOD value specified by the LOGON command that started a session, and rekeys the data into corresponding fields of the session's virtual terminal display buffer.

- A fully functional image of a virtual terminal screen may be displayed at a real terminal following any successful CA Verify for CICS session command, by simply displaying an ISPF panel that defines a dynamic area named PTEBUFF and attributes defined by the CA Verify for CICS LOGON command PANELDATA option.
- When a LOGON command used to start a CA Verify for CICS session and an ISPF panel used to display the session specify the same USERMOD value, all data entered by a terminal user while a virtual terminal screen image is displayed may be rekeyed into the session's virtual terminal display buffer by issuing a single TYPE command that specifies USERMOD.
- A REXX exec illustrated in ISPF Session Panels (see page 290) uses CA Verify for CICS session commands and an ISPF panel to start and manage interactive sessions with other online systems while using any ISPF application.

ENTER, PF1-24, PA1-3, PEN, CLEAR and ATTN

Keywords send data to an online system that is formatted like the data sent when a like-named key is pressed on a real terminal keyboard. The AID keyword identifies a variable that follows, &aid, which contains the name of the key to be pressed to send data. Virtual terminal keys that send data may be coded as the last TYPE command keystroke only.

A TYPE command that does not specify a key that sends data to an online system completes when the last specified keyboard operation has been performed.

When a key that sends data is pressed, the virtual terminal keyboard is locked and data identical to the data sent by a real terminal is sent to the online system. After sending data, the TYPE command normally waits for and receives response messages from the online system, until a message is received that frees the terminal keyboard, before returning to the exec that issued the command.

The HANDLE command may be issued to instruct the TYPE command to return immediately after sending data or to return after receiving any message even if the message does not free the keyboard. A TYPE command completes with a TIMEOUT condition if a response is not received within a time limit specified by a previous WAIT command.

The TYPE command returns updated information in session variables, as documented in CA Verify for CICS REXX Variables.

Conditions and Return Codes

At the completion of a TYPE command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes TYPE command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
LOCKOUT	4		Times out waiting for port lock
TIMEOUT	8		Timed out waiting for response
LOGOFF	16		Session ended
ENDTRACE	20	HALT	Trace abended (ie., SD37)
CANCEL	24	HALT	CANCEL command processed
FAULT	34	HALT	Invalid command or system error

UNLOCK

The UNLOCK command unlocks the port being accessed by an exec when the command is issued. A port may be unlocked only by an exec that is running on the same port as the exec that issued the LOCK or ACCESS command that locked the port.

If another exec task was delayed because it attempted to use resources anchored to the port being unlocked, the exec task is resumed when the port is unlocked.

Example:

```
" UNLOCK "
```

Format:

```
UNLOCK
```

Conditions and Return Codes

At the completion of an UNLOCK command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes UNLOCK command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
LOCKOUT	4		Timed out waiting for port lock
TIMEOUT	8		Time out waiting for response
LOGOFF	16		Session ended
ENDTRACE	20	HALT	Trace abended
CANCEL	24	HALT	CANCEL command processed
FAULT	64	HALT	Invalid command or system error

WAIT

The WAIT command waits for a specified period of time to elapse, or limits the amount of time that subsequent CA Verify for CICS commands wait for online system responses, data to be queued, or locks.

Examples:

```
" WAIT &mssth "

" WAIT  50000 on response "

" WAIT  forever on pull "
```

Format:

```
WAIT hhmmssth | FOREVER

( ON event )
```

Descriptions

hhmmssth is a time interval, expressed in hours (hh), minutes (mm), seconds (ss) and tenths and hundredths of seconds (th), or the keyword, FOREVER. FOREVER is valid only when ON event is specified.

ON event is optional. If not specified, the WAIT command simply waits for the specified interval to elapse. ON indicates that the time limit applies only when waiting for a named event to complete.

- ON LOCK limits the amount of time that CA Verify for CICS commands will wait for a resource lock, or the time that the LOCK command will wait for a port lock.
- ON PULL limits the amount of time that the CA Verify for CICS PULL command will wait for data when a PULL command is issued while the accessed data queue is empty.
- ON RESPONSE limits the amount of time that a CA Verify for CICS session command will wait for a required online system response to a virtual terminal input.

Conditions and Return Codes

At the completion of a WAIT command, the REXX variable RC indicates any CA Verify for CICS condition encountered while executing the command. The following table describes WAIT command return codes and default REXX conditions that may be raised for some CA Verify for CICS conditions.

Conditions	RC	Default Signal	Reason for Condition
	0		Command completed successfully
CANCEL	24	HALT	CANCEL command processed
FAULT	64	HALT	Invalid command or system error

Chapter 14: REXX Variables

This chapter documents special variables defined by CA Verify for CICS to provide information to a REXX exec.

CA Verify for CICS REXX Variables

CA Verify for CICS commands return specific information to execs in CA Verify for CICS defined REXX variables.

- All CA Verify for CICS defined REXX variable names begin with the characters PTE. Execs that use CA Verify for CICS commands for REXX should not create REXX variables that begin with PTE. Observing this convention ensures that there will be no conflict between user defined variables and current or future CA Verify for CICS defined variables.
- PTEPORT, PTEAPORT, PTEPORTS, PTEINFO and PTEMESS are used to return unique information and are always returned using the defined simple variable names.
- All other CA Verify for CICS defined variables are session variables, used by session commands to return information about a session on a CA Verify for CICS port accessed by an exec. If an exec accesses sessions on more than one port, session information is not unique. Session information is returned to an exec using either simple or compound session variable names that are unique for each session accessed by an exec.
- All variable names defined below are simple variable names. Session commands return information using simple session variable names by default or when an exec accesses the session on the exec's port by issuing an ACCESS command that instructs session commands to return information using simple session variable names.
- Compound session variable names have a REXX stem that is a defined simple session variable name and a tail that is a session's port number. For example, PTEBUFF.2 is a compound session variable name that may be used by session commands to return the session information defined for PTEBUFF when an exec accesses the session on port 2. Session commands return information using compound session variable names when an exec accesses a session on any port, including the session on the exec's port, by issuing an ACCESS command that instructs session commands to return information using compound session variable names.
- Session commands return session variables if a session is active on the accessed port at the completion of the session command. Session commands drop all session variables except PTERSENS if a session is not active on the accessed port at the completion of the command. The value of a dropped variable symbol is the value of the symbol itself.

PTEALARM

When a message is received from an online system, PTEALARM is set to YES if the message sounded the terminal alarm, or is set to NO if the message did not sound the terminal alarm.

PTEAPORT

Every ACCESS command sets PTEAPORT to the number of the port being accessed by the exec. ACCESS EXECPORT or ACCESS * accesses the port on which an exec is executing. ACCESS NEWPORT or an ACCESS command that specifies no operands assigns and accesses a previously unused port. ACCESS FREEPORT accesses a port that may have been previously used, but is not currently being used, or accesses a previously unused port. An ACCESS command may specify the number of the port to be accessed.

PTEASIZ

When a session is started, PTEASIZ is set to one of the values 24X80, 32X80, 43X80or 27X132, and indicates the alternate screen size established for the session.

PTEBUFF

PTEBUFF is a copy of the virtual terminal display buffer used for a CA Verify for CICS session. The size of the buffer, in rows and columns, is indicated by the session variables PTEROWS and PTECOLS. Screen rows are arranged serially in PTEBUFF, so that when PTECOLS is 80, screen row 2 begins at position 81 in PTEBUFF.

The format of the data and attributes stored in PTEBUFF may be specified by a QUERY SESSION command. In any other case, the format of PTEBUFF is determined by the BASICDATA, EXTDSDATA, PANELDATA, IMAGEDATA or ATTRIBUTES options specified by the LOGON command that started the session.

PTECC

A chain is a complete message sent from or to a terminal. PTECC is a count of the outbound chains received at a virtual terminal during the execution of the previous CA Verify for CICS session command. PTECC is set to 0 (zero) if no message is received during the execution of an INVITE or DELAY command.

PTECOL

A virtual terminal cursor works like a real terminal cursor. An online system may send orders that move the cursor to any display position. The TYPE command moves the cursor while keying data into a virtual terminal display, when tab or other keys are used, and moves the cursor to any specified display position. At the completion of any CA Verify for CICS session command, PTEROW and PTECOL identify the current cursor row and column in the virtual terminal display buffer.

PTECOLS

PTECOLS contains the current number of virtual terminal screen columns. The possible values of PTECOLS are fixed when default and alternate screen sizes are established during session initiation. The current value of PTECOLS at any point in time is controlled by the online system using 3270 ERASE-WRITE or ERASE-WRITE-ALTERNATE commands.

PTECSRP

During the execution of a TYPE command, the cursor is advanced as data is keyed into a virtual terminal screen. When an online system sends a terminal message, the message may reposition the cursor. Whenever the cursor is repositioned, PTECSRP is set to the position of the cursor within PTEBUFF. When the cursor is at row 1, column 1, PTECSRP is 1. When the cursor is at row 2, column 1 of an 80-column screen, PTECSRP is 81.

PTEDSIZ

When a session is started, PTEDSIZ is set to one of the values 24X80, 32X80, 43X80or 27X132, and indicates the default screen size established for the session.

PTEEAB

PTEEAB is an image of a virtual terminal Extended Attribute Buffer used to store 3270 extended attribute data that correspond to the field attributes and data stored in PTEBUFF. Refer to 3270 Attribute Values for a description of the data contained in an extended attribute buffer.

When session variables are returned by a QUERY SESSION command, PTEEAB is returned if EXTDSDATA is specified, and is not returned if BASICDATA, PANELDATA or IMAGEDATA are specified. In any other case, when session variables are returned by a session command, PTEEAB is returned if the LOGON command that started the session specified EXTDSDATA or if the LOGON command started an EXTDS session and did not specify BASICDATA, PANELDATA or IMAGEDATA.

PTEINFO

If the QUERY command determines that a named CA Verify for CICS resource is active, the resource name is returned in PTEINFO. If the QUERY command determines that a named CA Verify for CICS resource is not active, the concatenation of NO and the resource name is returned in PTEINFO. For example, if the command QUERY SESSIONS finds a session active on any CA Verify for CICS port, SESSIONS is returned in PTEINFO, otherwise, NOSESSIONS is returned in PTEINFO.

PTEKB

PTEKB is the state of a virtual terminal keyboard and is one of the following values:

- BUSY means that data has been typed but not entered and that additional data may be typed.
- FREE means that data may be typed, but no data has been typed since the keyboard was freed.
- INHIBITED means that a session is timed out, outbound or in another state that does not permit data to be typed, but a TYPE ATTN command is permitted.
- LOCKED means that a session has timed out in a state that does not permit the use of the TYPE statement.

  Note: Keyboard state names that have lower alphabetic values are "more free" and names with higher values are locked tighter. Data may be typed when PTEKB is LE "FREE", but may not be typed when PTEKB is GE "INHIBITED".

PTELUDS

When a session is started, PTELUDS is set to one of the values BASICDS or EXTDS, and indicates the 3270 data stream level supported by the virtual terminal used for the session.

PTEMESS

An error or information message may be stored in PTEMESS during the execution of any CA Verify for CICS command. Selected messages are sent to the terminal or to another message destination automatically, if a MONITOR command has been issued that specified SIGNALS, CONDITIONS or the PTEMESS variable name.

PTEPLU

When a session is started with any Primary Logical Unit (PLU), the network name of the online system is stored in PTEPLU. While most sessions are started by the LOGON command, sessions may also be started while executing any session command, if an online system passes the PLU end of a session to another PLU. All session commands store the name of the current PLU in PTEPLU if a session is passed to another PLU.

PTEPORT

All ACCESS commands store the number of the port on which an exec is executing in PTEPORT.

PTEPORTS

A QUERY command may specify one of the keywords, SESSIONS, TRACES, TASKS or QUEUES, to obtain a list of the ports on which the named CA Verify for CICS resource exists. For example, an exec may issue a QUERY SESSIONS command to set PTEPORTS to a list of port numbers on which CA Verify for CICS sessions are active. If sessions are active on ports 3, 4, and 7, the value 3 4 7 is stored in PTEPORTS.

PTEROW

A virtual terminal cursor works like a real terminal cursor. An online system may send orders that move the cursor to any display position. The TYPE command moves the cursor while keying data into a virtual terminal display, when tab or other keys are used, and moves the cursor to any specified display position. At the completion of any CA Verify for CICS session command, PTEROW and PTECOL identify the current cursor row and column in the virtual terminal display buffer.

PTEROWS

PTEROWS is the current number of virtual terminal screen rows. The possible values of PTEROWS are fixed when default and alternate screen sizes are established during session initiation. The current value of PTEROWS at any point in time is controlled by the online system using 3270 ERASE-WRITE and ERASE-WRITE-ALTERNATE commands.

PTERSENS

If VTAM is unable to start a session requested by a LOGON command, or if an active session fails, or if data sent to an online system is rejected during the execution of a CA Verify for CICS session command, the eight character sense code received from VTAM or the online system is stored in PTERSENS. Sense codes are defined by SNA and indicate the cause of a session failure or other negative response. If a LOGON fails, or a session fails unexpectedly, an exec may analyze the contents of PTERSENS to determine the cause. For example, 08570002 means the PLU is not active and 087D0001 indicates a session services path error, possibly resulting from an attempt to start a session with an undefined PLU. Unlike other session variables, PTERSENS is not dropped at the end of a session. General information on sense codes can be found in SNA Sense Codes.

PTERTIME

When a message is received from an online system, response time is calculated by subtracting the time of the previous terminal input from the time the message is received, and the calculated response time, expressed in hundredths of a second, is stored in PTERTIME.

PTESENSE

When CA Verify for CICS sends a negative response to a message received from an online system, the four-character SNA sense code sent with the negative response is stored in PTESENSE. The sense code indicates the reason that the online system request was rejected. If a negative response is not sent during the execution of a CA Verify for CICS session command, the value '0000' is stored in PTESENSE. General information on sense codes can be found in SNA Sense Codes.

PTESIZE

When a CA Verify for CICS session is started or when a 3270 ERASE-WRITE command is received from an online system, the default screen size, calculated as the default screen size rows times the default screen size columns, is stored in PTESIZE. When a 3270 ERASE-WRITE-ALTERNATE command is received, the alternate screen size is calculated as the alternate screen size rows times the alternate screen size columns, is stored in PTESIZE.

PTESLU

When a session is started by a LOGON command, the name of the VTAM APPL used as a CA Verify for CICS virtual terminal is stored in PTESLU. SLU is an acronym for Secondary Logical Unit. The name is either a name supplied by the exec, or is an assigned name, if a name was not supplied by the exec.

PTESTATE

PTESTATE is a name that describes the state of a CA Verify for CICS session. PTESTATE may be any state name described in CA Verify for CICS Logical Unit States, except OFFLINE. When a session ends, instead of storing the value OFFLINE, PTESTATE is dropped.

PTETTIME

When a TYPE command is used to send terminal input to an online system, think time is calculated by subtracting the time at which the previous message was received from the online system from the time of the terminal input, and the calculated think time, expressed in hundredths of a second, is stored in PTETTIME.

Chapter 15: Installation Verification Procedure and Demo Session

This section contains the following topics:

Overview

The best way to learn how to take advantage of CA Verify for CICS is to use the demo session. When you complete this session, you'll understand how to use CA Verify for CICS to test application program changes.

The Sample Program

A sample application program is distributed and automatically installed with CA Verify for CICS. This simple program is an order processing application for a fictional company, Carol's Cookies Company. It allows a telephone operator to process new orders for cookies, as well as check the current status of existing orders. Modifications have been made to "Release 6.2" of the order entry program in preparation for "Release 6.3."

Three versions of the demo program are installed, including:

6.2

The original version *before* any changes (Release 6.2)

6.3

The changed version with an error (Release 6.3)

6.3 FIX

The corrected version of the modified program

The Demo

In the demo that follows, you will use CA Verify for CICS to test whether the output generated by Release 6.3 matches that of Release 6.2 with no errors or unexpected changes.

Follow these steps:

1. **Log a test stream.** You'll start CA Verify for CICS and specify testing options, and then log the input and output screens associated with Release 6.2 of Carol's Cookies Company order processing program.
2. **Create Rules.** Next, you'll examine Release 6.3 of the order entry program and create rules to tell CA Verify for CICS how to handle the expected differences between the 6.2 and 6.3 releases.
3. **Test Release 6.3.** Then, you will run your logged test stream on Release 6.3. There is a deliberate error in Release 6.3 that CA Verify for CICS will identify when it compares the logged output screens from Step 1 with the new output screens produced during the run.
4. **Debug 6.3 and test the final program**. Next, you will correct the demo program and retest. To simplify this demonstration, you will not actually need to debug the program. You will use **6.3 FIX** instead, which is the corrected version of Release 6.3.

The demo is divided into four parts that correspond to the previous steps. At the end of each part, you can continue on to the next part or stop and return later to where you left off.

Note: The defaults at your installation may differ from the delivered defaults, so some of your menus may differ slightly from the menus presented here. These differences won't affect your use of the demo session.

Installation Procedure for Release 6.2

Before starting the demo, check to see whether you have Release 6.2 of Carol's Cookies Company installed. This can be done by typing XCCC on a clear screen and checking the release number that appears in the upper right corner. If 6.2 appears, then you have Release 6.2 installed; if 6.2 does not appear, then you must install Release 6.2.

To install Release 6.2, type the command shown following from a cleared screen.

```
install 6.2
```

When the following message is displayed, the installation has completed successfully.

CAROL'S COOKIES COMPANY ORDER SYSTEM RELEASE 6.2 HAS BEEN INSTALLED.

Clear the screen and you are now ready to complete the tasks in Part 1: Logging a Test Stream.

Part 1: Log a Test Stream

The first part of the demo involves logging a test stream for Release 6.2. You will use this test stream throughout the rest of this demo.

Follow these steps:

1. Start a logging session.
2. Process a transaction with Release 6.2 of the Carol's Cookies Company program.
3. Exit the demo program and stop logging.

Each of the previous steps involves sub-tasks that must be followed, in the order they appear, for that step to be completed.

At the end of Part 1, you will have created a Release 6.2 test stream called tcads.ccc.orderapp.001.

Step 1. Start the Logging Session

To begin logging a test stream, invoke CA Verify for CICS and specify logging options for this session. You will need to complete the following tasks:

A. Invoke CA Verify for CICS.

B. Select the logging option from the Primary Options Menu.

C. Specify logging options for this session.

A. Invoke CA Verify for CICS

Action:

Clear your screen. Type xtca and press Enter.

Result:

CA Verify for CICS displays the Primary Options Menu.

B. Select from the Primary Options Menu

The Primary Options Menu lists all of CA Verify for CICS's primary functions. Your first task is to log the screens associated with the demo program.

Action:

Type L and press Enter.

```
VERSION x.x.x -------  CA VERIFY PRIMARY OPTIONS MENU  ----------------13:44:59
ENTER COMMAND ===>  L                                                TERM: 60L2050
                                                                     OPER:
   L  LOG A TEST STREAM

   B  BROWSE A TEST STREAM

   R  RUN A TEST STREAM

   E  EDIT A TEST STREAM

   M  MAINTAIN RULES

   I  INQUIRY/TERMINATION OF FUNCTIONS

   U  UTILITIES

   T  TUTORIAL

   X  EXIT
         *** ENTER "KEYS" COMMAND TO SEE PF KEY ASSIGNMENTS ***


COPYRIGHT © 2012 CA.   ALL RIGHTS RESERVED
```

Result:

CA Verify for CICS displays the Log Options Menu.

C. Specify Logging Options for This Session

The Log Options Menu lets you specify whether screens will be logged from your terminal, another terminal, or multiple terminals.

Action:

Type t to capture screens from your terminal only, and press Enter.

```
---------------------------  LOG OPTIONS MENU  ----------------------13:45:12
 ENTER COMMAND ===>  t

    T  THIS TERMINAL

    O  ANOTHER TERMINAL

    M  MULTIPLE TERMINALS

 ENTER ONE OF THE ABOVE OPTIONS AND PRESS ENTER TO CONTINUE OR F3 TO END
```

Result:

CA Verify for CICS displays the Single Terminal Log menu.

Note: You could have bypassed the Log Options Menu by typing L.T on the Primary Options Menu to go directly to the Single Terminal Log Menu.

Action:

Type the following information on the Single Terminal Log Options menu. This menu lets you specify the name for the series of screens — test stream — you are going to log, and options that affect the logging.

The DDNAME, APPLICATION, MEMBER, and VERSION information identify the test stream.

- The DDname (FILENAME for VSE) is the CA Verify data set into which the screens will be logged. Your installation may have many valid DDnames. TCADS is the default and should be available. However, you can tab to this field and overtype it if necessary.

- Application and Member are the names you specify for the test stream.
  - Tab to the Application field and Type ccc.
  - Then tab to the Member field and Type orderapp.
- The Version identifies the test stream when multiple test streams have the same DDname, Application, and Member names. Leave 001 as the default.

DESCRIPTION

Describes the test stream. Tab to this field and type up to three lines of any meaningful text.

STOP OPTION

Specifies how you will terminate logging. MAN is the default and indicates that you will stop logging manually. You can also specify a PF key, a PA key, or CLEAR. Leave MAN as the default.

PROCESS WITH RULES

Lets you apply a ruleset to this logging session. If Y is typed in this field, a list of available rulesets will appear when you press Enter. These rulesets contain rules that specify user IDs, terminals, or transaction IDs that are to be included or excluded from the test stream. Since you do not wish to apply a ruleset to this logging session, leave N as the default.

TEST STREAM PROTECTION

Limits access to this test stream by other users. Leave this field blank.

LOG INPUT SCREENS ONLY

Logs only input screens from the terminal to the application. Leave N as the default.

EXTEND TEST STREAM

Specifies whether or not this test stream is to be appended to an existing test stream. Leave N as the default.

RULESET NAME

Contains the name of the ruleset to be used during the logging process, if one was selected. This field will be blank.

When your screen is completed like the one shown next, press Enter.

```
-------------------------  SINGLE TERMINAL LOG  ----------------------13:45:18
ENTER COMMAND ===>                                                        L1

LOG TEST STREAM AS:
  DDNAME      ===> TCADS
  APPLICATION ===> ccc
  MEMBER      ===> orderapp
  VERSION     ===> 001

DESCRIPTION ===>   carol's cookies company demo program
            ===>
            ===>

STOP OPTION                    ===>  MAN  (MAN, PF__, PA_, OR CLEAR)
PROCESS WITH RULES             ===> N    (Y/N)
TEST STREAM PROTECTION         ===>      (R-READ W-WRITE P-PRINT)
LOG INPUT SCREENS ONLY         ===> N    (Y/N)
EXTEND TEST STREAM             ===> N    (Y/N)


RULESET NAME:


F1-HELP     F3-END     F4-RETURN
```

Result:

CA Verify for CICS clears your screen.

Step 2. Process a Transaction

When you have completed Step 1, CA Verify for CICS will begin logging every input and output screen until you stop logging. Now you can invoke the Carol's Cookies Company program, Release 6.2. To complete Step 2, you will need to complete the following tasks:

A. Start Release 6.2.

B. Run a sample transaction. CA Verify for CICS will be recording your keystrokes and keeping screen shots as you process this transaction.

A. Start the Carol's Cookie Company Program Release 6.2

Start the current version of the Carol's Cookies Company order entry program.

Action:

Clear your screen (if you are continuing directly from Step 1, your screen will already be clear). Type **xccc** and press Enter.

Result:

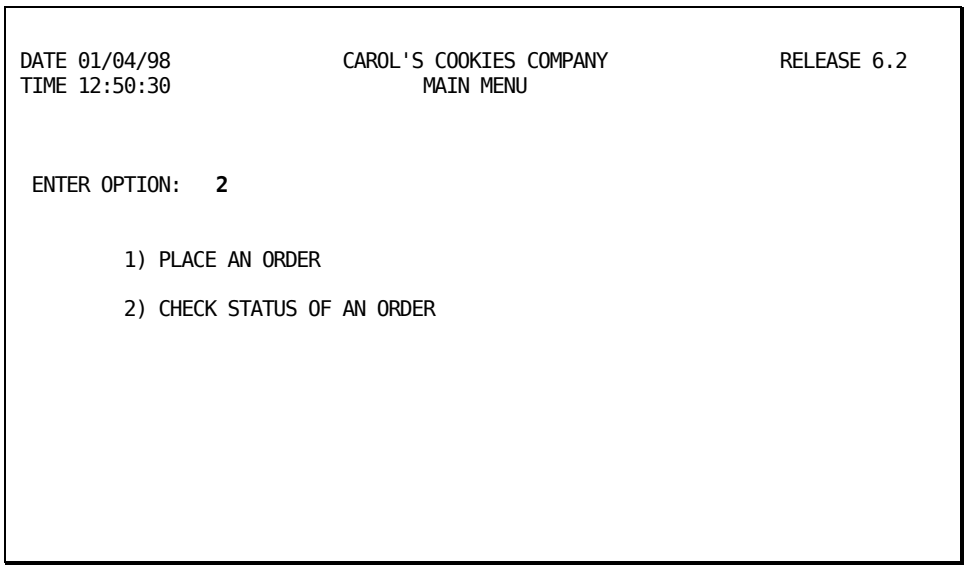
You will go to the program's Main Menu.

B. Check the Status of a Cookie Order

Process a transaction as if a customer called in to check the status of his order.

Action:

Type **2** in the Select Option field, as illustrated next. Press Enter.



Result:

The program's Order Status Selection screen is displayed.

Action:

Type **s** next to the name, John Smith, whose order you are going to check, and press **Enter**.

```
DATE 01/04/98              CAROL'S COOKIES COMPANY          RELEASE 6.2
TIME 12:49:58               ORDER STATUS SELECTION




          CUSTOMER NAME      ORDER NUMBER   STATUS       AMOUNT

      s   JOHN SMITH          100-120-15    ON ORDER      17.90
      _   BILL JONES          100-205-12    SHIPPED       22.67
      _   SUE WILLIAMS        100-271-15    ON ORDER      34.30
```

Result:

The status and details of this customer's order are displayed on the Order Status screen as shown next.

```
DATE 01/04/98              CAROL'S COOKIES COMPANY          RELEASE 6.2
TIME 12:50:06                    ORDER STATUS




               CUSTOMER NAME: JOHN SMITH

               STATUS: O N ORDER


               ITEM # DESCRIPTION  QUANTITY    PRICE   AMOUNT
                137   CHOC CHIP      1 DZ       5.00     5.00
                474   OATMEAL RSN    2 DZ       5.00    10.00

               SUBTOTAL:                 15.00
               TAX:                        .90
               SHIPPING & HANDLING:       2.00

               TOTAL:                    17.90
```

You have completed the transaction. Return to the Main Menu.

Action:

Press PF3 two times.

Result:

You return to the program's Main Menu.

Step 3. Conclude this Logging Session

In Step 2, you completed a transaction and returned to the Main Menu. Now, complete these tasks:

A. Exit the demo program.

B. Stop logging.

C. Review the summary information for this logging session.

A. Exit the Carol's Cookies Company Program

Action:

From the Main Menu, press PF3 to exit the Carol's Cookies Company order entry program, Release 6.2.

Result:

You are returned to CICS and your screen is cleared.

B. Stop Logging This Test Stream

You are now ready to stop logging.

Action:

Type xtca stop and press Enter.

Result:

CA Verify for CICS stops logging and displays the Log Termination menu.

C. View Log Termination Information

The Log Termination menu displays statistics for the test stream you just logged.

Action:

Check the statistics and compare yours to the statistics shown next. Press PF3 to exit from the Log Termination screen.

```
CCC.ORDERAPP.001 ----------  LOG TERMINATION  -----------------------13:46:45
ENTER COMMAND ===>                                                       L5

 DESCRIPTION: CAROL'S COOKIES COMPANY DEMO PROGRAM



            LOG:       RUN:       EDIT:         TEST STREAM:        IN:   OUT:
 INVOKED BY:                                     TOTAL SCREENS:       6      6
 INVOKED ON: 02/11/1998                          AVERAGE BYTES:      47    318
 START TIME: 13:46:12
 DURATION:   00:08:39   00:00:01
 SYSTEM:     GRIA5451
 STATUS:     NORMAL
 TERMINAL:   A60L2050
 VSAM CI'S:  1

 AVERAGE THINK TIME:    00:01:26.239
 AVERAGE RESPONSE TIME: 00:00:00.198
 MAXIMUM SCREEN SIZE:   24 BY  80
 PROTECTION STATUS:                                  OWNER:
 ORIGINATING TEST STREAM:                CREATED BY FUNCTION: LOG

F1-HELP     F3-END     F4-RETURN
```

Result:

CA Verify for CICS displays the Primary Options Menu.

Note: This is the end of Part 1. You can continue to Part 2 or stop the demo at this time. To return to this menu at a later time, follow the directions in the section *Invoke* CA Verify for CICS in Part 1.

Action:

Type x in the command line and press Enter.

Result:

You exit CA Verify for CICS and return to CICS.

Part 2: Install Release 6.3 and Create Rules for Expected Changes

Install Release 6.3 and Restart Product

Now that you have finished logging the screens in Release 6.2, you are ready to install Release 6.3 of Carol's Cookies Company.

Action:

```
Type install 6.3 on a clear screen under CICS and press Enter.
install 6.3
```

Result:

Your screen will display the message:

CAROL'S COOKIES COMPANY ORDER SYSTEM RELEASE 6.3 HAS BEEN INSTALLED

Start CA Verify for CICS.

Action:

Clear the screen. Type xtca. Press Enter.

Result:

The Primary Options menu opens.

The rest of Part 2 of the demo involves creating rules using the Rules function. Certain changes are expected in Release 6.3, and you do not need CA Verify for CICS to flag them while running your test stream. With the Rules function, you identify these differences and specify how CA Verify for CICS is to handle them. For more information, see the chapter Rules Function in this guide or the *Rules Primer*.

Create Rules for Expected Changes

Follow these steps:

1. Add a Ruleset for the test stream you logged in Step 1.
2. Create a rule for the expected changes to the Main Menu.
3. Create another rule for the Order Status screen.
4. Exit CA Verify for CICS.

Each of the previous steps involve sub-tasks that must be followed for the step to be completed.

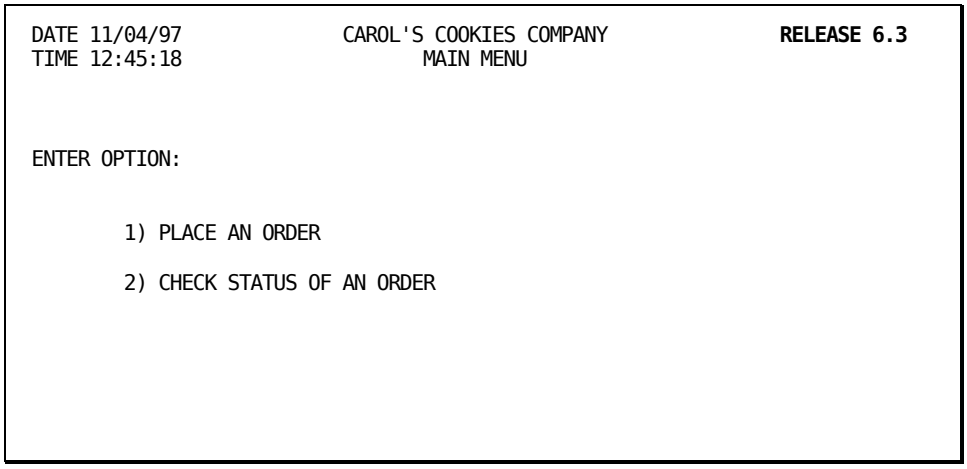
At the end of this section, you will have created a set of rules to be used when testing 6.3 in Part 3 and 6.3 FIX in Part 4.

Expected Changes in Release 6.3

Let's review the changes that will be coming in Release 6.3 of this application and look at some examples of the new screens.

1. The date and time will now be variable fields.
2. The release level will be changed from 6.2 to 6.3.

You will need to create rules to notify CA Verify for CICS of these changes.



Step 1. Add a Ruleset for Your Test Stream

You first need to create a ruleset that contains the rules that will handle the changes from Release 6.2 to Release 6.3. You must complete these tasks:

A. Select the Maintain Rules function from the Primary Options Menu.

B. Add a ruleset.

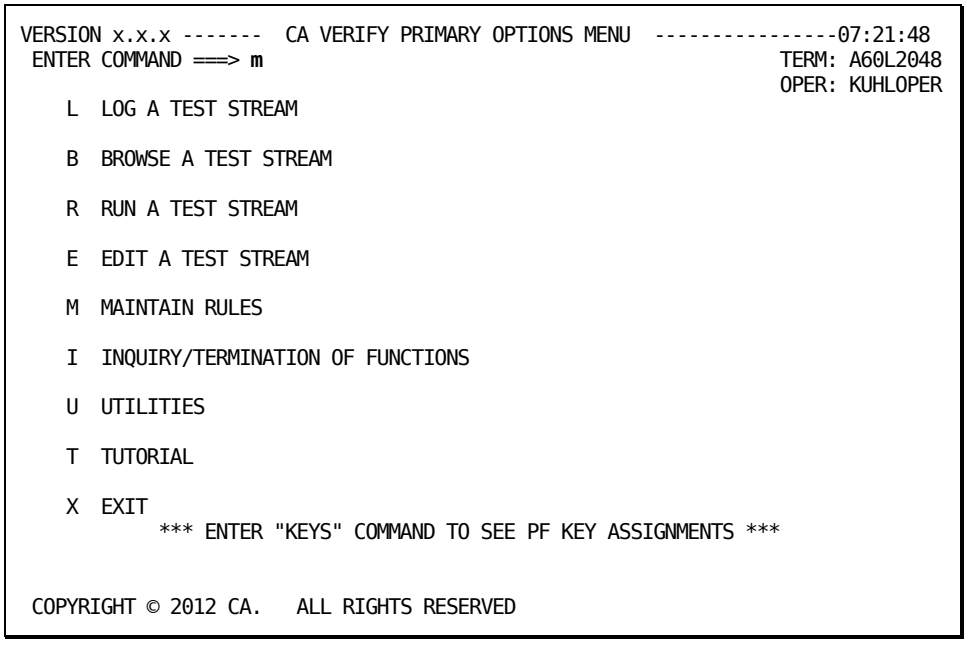
C. Add a ruleset description.

A. Select the Maintain Rules Function

At the end of Part 1, you exited Release 6.2 of the Carol's Cookies Company order entry program, stopped logging, and returned to the Primary Options Menu. From here, choose the option Maintain Rules.

Action:

Type m in the command line and press Enter.



Result:

CA Verify for CICS displays the Maintain Rules menu.

B. Add a Ruleset

On the Maintain Rules menu, you name the ruleset and select option A to add a ruleset. The name of the ruleset that you create must match the name of your test stream:

```
tcads.ccc.orderapp.001
```

Action:

Type a on the command line. Type ccc in the Application field. Type orderapp in the Member field. Press Enter.

```
--------------------------- MAINTAIN RULES --------------------------12:05:38
ENTER COMMAND ===> a

   A  ADD A RULESET                C  COPY A RULESET
   B  BROWSE A RULESET             D  DELETE A RULESET
   E  EDIT A RULESET               R  RENAME A RULESET

RULESET NAME:

  DDNAME      ===> TCADS
  APPLICATION ===> ccc            (LEAVE APPLICATION, MEMBER,
  MEMBER      ===> orderapp         OR VERSION BLANK AND PRESS
  VERSION     ===> 001              ENTER FOR A SELECTION LIST)




 F1-HELP      F3-END       F4-RETURN
```

Result:

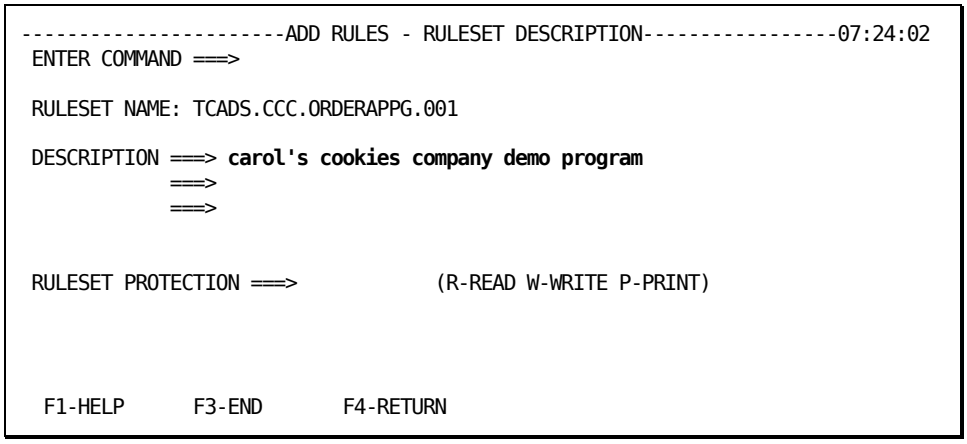
The ruleset you named is created, and CA Verify for CICS displays the Add Rules—Ruleset Description panel.

C. Add a Ruleset Description

On the Add Rules—Ruleset Description panel, the test stream description is automatically filled in.

Action:

Press Enter.



Result:

CA Verify for CICS displays the Add Rules—Rule Actions panel.

Step 2. Create the Rules for the 6.3 Changes

You have just given a name to the set of rules you are creating in this demo. You can now create the rules for this ruleset. These rules will handle the expected changes for Release 6.3 of Carol's Cookies Company. To create the rules, you must complete these tasks:

A. Assign a name to the rule.

B. Select a model screen.

C. Set up recognition criteria.

D. Change the release level from 6.2 to 6.3.

E. Set up the date and time fields as variables.

A. Assign a Name to the Rule

CA Verify for CICS automatically creates a rule name. You can change it if you wish, but the generated name is used for the purposes of this demo.

Action:

RUL00001 appears in the Rule Name field. In the Description field, type release 6.2 to 6.3 changes. Press Enter.

```
----------------------ADD RULES - RULE ACTIONS---------------
 ENTER COMMAND ===>

RULE NAME: RUL00001            RULESET NAME:     TCADS.CCC.ORDERAPP.001
                              TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

DESCRIPTION  ===> release 6.2 to 6.3 changes
             ===>

       1  FIELD RECOGNITION                   11  GENERATE FIELD VALUE
       2  SCREEN RECOGNITION                  12  INSERT SCREENS
       3  VARIABLE FIELD                      13  DELETE SCREENS
       4  DELETE FIELD                        14  CUT SCREEN FIELD
       5  MOVE FIELD                          15  PASTE SCREEN FIELD
       6  CHANGE FIELD VALUE                  20  USERID LOGGING
       7  NEW FIELD                           21  TERMINAL ID LOGGING
       8  CHANGE AID KEY                      22  TRANSACTION ID LOGGING
       9  CHANGE CURSOR LOCATION
      10  CHANGE WCC VALUES


  F1-HELP      F2-PREVIEW      F3-END      F4-RETURN      F9-SELECT
```

Result:

CA Verify for CICS redisplays the Add Rules—Rule Actions panel with this message:

TCA5155 - RULE IS NOT YET COMPLETE -- MUST ADD RULE ACTION.

B. Select a Model Screen

In this demo, you will be using models to obtain screen coordinates and other information for the fields you are changing.

Action:

Press PF9 to select a model screen.

```
CCC.ORDERAPP.001   -------  RULES: RECORD SELECTION  ------------------14:55:24

ENTER COMMAND ===>                                                        B7

   TERMINAL  TRAN  T/R TIME   OP    AID    VIEW   RECORD  ROW: 1 COL: 30
_  A60L2048  XCCC  00:41.690  RM    ENTER            1
_  A60L2048  XCCC  00:00.156  EW                     2   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  04:39.785  RM    ENTER            3   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  00:00.900  EW                     4   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  00:32.297  RM    ENTER            5   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  00:00.115  EW                     6   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  01:14.403  RM    PF3              7   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  00:00.005  EW                     8   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  00:01.098  RM    PF3              9   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  00:00.005  EW                    10   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  01:28.161  RM    PF3             11   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  00:00.005  EW                    12
***  END OF RECORDS  ***









 TYPE AN "S" TO SELECT A RECORD
F1-HELP   F3-END   F4-RETURN   F7-UP   F8-DOWN   F9-FORMAT   F10-LEFT   F11-IGHT
```

Result:

The Rules: Record Selection panel is displayed.

The Rules—Record Selection panel shows the records as they currently exist in the test stream. From the records listed on this panel, locate the one you want to use as your model screen and select it.

Action:

Tab to record 2 and type **s** on the line to the left. Press Enter.

```
        --------- RULES - RECORD SELECTION --------------12:00:30
ENTER COMMAND ===>                                                    M7

   TERMINAL  TRAN  T/R TIME   OP    AID    VIEW   RECORD  ROW: 1 COL: 30
_  A60L2048  XCCC  00:41.690  RM    ENTER             1
s  A60L2048  XCCC  00:00.156  EW                      2   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  04:39.785  RM    ENTER             3   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  00:00.900  EW                      4   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  00:32.297  RM    ENTER             5   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  00:00.115  EW                      6   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  01:14.403  RM    PF3               7   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  00:00.005  EW                      8   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  00:01.098  RM    PF3               9   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  00:00.005  EW                     10   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  01:28.161  RM    PF3              11   CAROL'S COOKIES COMPA
_  A60L2048  XCCC  00:00.005  EW                     12
***  END OF RECORDS  ***




TYPE AN "S" TO SELECT A RECORD
F1-HELP  F3-END  F4-RETURN  F7-UP  F8-DOWN  F9-FORMAT  F10-LEFT F11-RIGHT
```

Right:

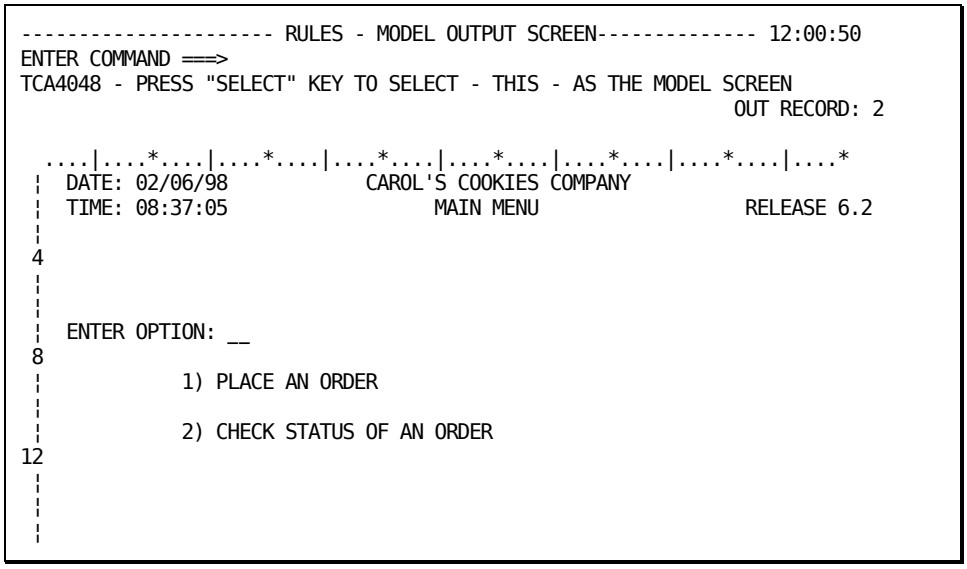
CA Verify for CICS displays the Rules: Model Output Screen containing the record you selected.

Confirm Your Selection

You have selected the Main Menu to use as your model screen. Notice that you can scroll up, down, to the left and to the right to view all parts of the logical screen. When you are convinced that this is the correct screen, press PF9 (Select) to confirm your selection.

Action:

Press PF9.



Result:

You selection is confirmed and you are returned to the Add Rules—Rule Actions screen.

C. Specify Recognition Criteria

Now you need to specify the recognition criteria CA Verify for CICS will use to identify screens in this application. Use this criteria:

The "Carol's Cookies Company" title

Action:

Type a 1 in the Enter Command field and press Enter.

```
----------------------ADD RULES - RULE ACTIONS---------------
 ENTER COMMAND ===> 1

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                             TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

DESCRIPTION  ===> RELEASE 6.2 TO 6.3 CHANGES
             ===>

       1  FIELD RECOGNITION                  11  GENERATE FIELD VALUE
       2  SCREEN RECOGNITION                 12  INSERT SCREENS
       3  VARIABLE FIELD                     13  DELETE SCREENS
       4  DELETE FIELD                       14  CUT SCREEN FIELD
       5  MOVE FIELD                         15  PASTE SCREEN FIELD
       6  CHANGE FIELD VALUE                 20  USERID LOGGING
       7  NEW FIELD                          21  TERMINAL ID LOGGING
       8  CHANGE AID KEY                     22  TRANSACTION ID LOGGING
       9  CHANGE CURSOR LOCATION
      10  CHANGE WCC VALUES


  F1-HELP      F2-PREVIEW      F3-END      F4-RETURN      F9-SELECT
```

Result:

The Rules: Expected Screen will be displayed so you can cursor select the identification information. This screen will only appear if you have previously selected a model (as we did in the previous step). If no model screen is selected, the Add Rule Actions—Recognition Criteria screen will be displayed.

Place your cursor on a field to select it, and then press PF9 to drag that field information into the Recognition Criteria panel. CA Verify for CICS will do the work of figuring out the Row, Column, Length, Operator, and Value coordinates.

Action:

Position the cursor on the first **C** in Carol's Cookies Company on the top line, and then press PF9.

```
---------------------- RULES : EXPECTED SCREEN-------------- 12:00:50
ENTER COMMAND ===>
TCA4049 - MOVE CURSOR TO IDENTIFICATION CRITERIA FIELD; PRESS "SELECT" KEY
                                                          OUT RECORD: 2

 ....|....*....|....*....|....*....|....*....|....*....|....*....|....*....|....*
 ¦  DATE: 02/06/98           CAROL'S COOKIES COMPANY
 ¦  TIME: 08:37:05                MAIN MENU                  RELEASE 6.2
 ¦
 4
 ¦
 ¦
 ¦  ENTER OPTION: __
 8
 ¦           1) PLACE AN ORDER
 ¦
 ¦           2) CHECK STATUS OF AN ORDER
12
 ¦
 ¦
 ¦
```

Result:

The Add Rule Actions—Recognition Criteria screen will be displayed.

The information for the recognition criteria is entered automatically as shown in the following screen.

```
--------------ADD RULE  ACTIONS - RECOGNITION CRITERIA---------  ADD SUCCESSFUL
 ENTER COMMAND ===>

RULE NAME: RUL00001            RULESET NAME:     TCADS.CCC.ORDERAPP.001
                              TEST STREAM NAME: TCADS.CCC.ORDERAPP.001


 ROW COL  LEN  OPER VALUE
 01  030  023  EQ   CAROL'S COOKIES COMPANY_____________________________
                   ____________________________________________________
                   ____________________________________________________




  F1-HELP     F2-PREVIEW     F3-END    F4-RETURN     F9-SELECT
```

Action:

Press PF3.

Result:

You return to the Add Rules - Rules Actions screen.

D. Change the Release Level

You will now create a rule that will inform CA Verify for CICS that the release level has changed from 6.2 to 6.3 for this application

Action:

Type 6 in the command line and press Enter.

```
----------------------ADD RULES - RULE ACTIONS---------------
 ENTER COMMAND ===> 6

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                             TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

DESCRIPTION  ===> RELEASE 6.2 TO 6.3 CHANGES
             ===>

       1  FIELD RECOGNITION                   11  GENERATE FIELD VALUE
       2  SCREEN RECOGNITION                  12  INSERT SCREENS
       3  VARIABLE FIELD                      13  DELETE SCREENS
       4  DELETE FIELD                        14  CUT SCREEN FIELD
       5  MOVE FIELD                          15  PASTE SCREEN FIELD
       6  CHANGE FIELD VALUE                  20  USERID LOGGING
       7  NEW FIELD                           21  TERMINAL ID LOGGING
       8  CHANGE AID KEY                      22  TRANSACTION ID LOGGING
       9  CHANGE CURSOR LOCATION
      10  CHANGE WCC VALUES


  F1-HELP      F2-PREVIEW      F3-END      F4-RETURN      F9-SELECT
```

Result:

You will go to the Rules: Expected Screen panel where you will identify what you want to change using the model screen to obtain and type the location information for the field to be changed.

Indicate the field whose value has changed.

Action:

Position the cursor on the **2** of RELEASE 6.2 and press PF9 to drag the location information back to the Add Rule Actions—Change Field Value panel.

```
---------------------- RULES : EXPECTED SCREEN-------------- 12:00:50
ENTER COMMAND ===>
TCA4042 - MOVE CURSOR TO CHANGED FIELD AND PRESS "SELECT" KEY
                                                          OUT RECORD: 2

 ....|....*....|....*....|....*....|....*....|....*....|....*....|....*
 ¦  DATE: 02/06/98           CAROL'S COOKIES COMPANY
 ¦  TIME: 08:37:05                MAIN MENU                 RELEASE  6.2
 ¦
 4
 ¦
 ¦
 ¦  ENTER OPTION: __
 8
 ¦           1) PLACE AN ORDER
 ¦
 ¦           2) CHECK STATUS OF AN ORDER
12
 ¦
 ¦
 ¦
```

Result:

The information will be entered in the Add Rule Actions—Change Field Value panel.

Change the field value.

Action:

Type the new value, 3, over the old value, 2. Check that all other information is typed as it appears next. Press Enter to update, press PF3.

```
--------------------ADD RULE ACTIONS - CHANGE FIELD VALUE-----------08:23:24
 ENTER COMMAND ===>

RULE NAME: RUL00001            RULESET NAME:     TCADS.CCC.ORDERAPP.001
                              TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 CHANGED FIELDS:
 ROW COL  LEN  ATTR FIELD VALUE
 01  074  001  ____ 3_____________________________________________________
                   _____________________________________________________
                   _____________________________________________________

OUTPUT/INPUT FIELD  ===> O      (O=OUTPUT, I=INPUT)



 F1-HELP  F2-PREVIEW  F3-END  F4-RETURN  F9-SELECT
```

Result:

The field change information is saved. You are returned to the Add Rules—Rule Actions menu.

F. Set Up the Date and Time Fields as Variables

Next, make the date and time fields into variable fields.

Action:

Type 3 in the command line and press Enter.

```
----------------------ADD RULES - RULE ACTIONS---------------
 ENTER COMMAND ===> 3

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                             TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

DESCRIPTION  ===> RELEASE 6.2 TO 6.3 CHANGES
             ===>

       1  FIELD RECOGNITION                   11  GENERATE FIELD VALUE
       2  SCREEN RECOGNITION                  12  INSERT SCREENS
       3  VARIABLE FIELD                      13  DELETE SCREENS
       4  DELETE FIELD                        14  CUT SCREEN FIELD
       5  MOVE FIELD                          15  PASTE SCREEN FIELD
       6  CHANGE FIELD VALUE                  20  USERID LOGGING
       7  NEW FIELD                           21  TERMINAL ID LOGGING
       8  CHANGE AID KEY                      22  TRANSACTION ID LOGGING
       9  CHANGE CURSOR LOCATION
      10  CHANGE WCC VALUES


  F1-HELP      F2-PREVIEW      F3-END      F4-RETURN      F9-SELECT
```

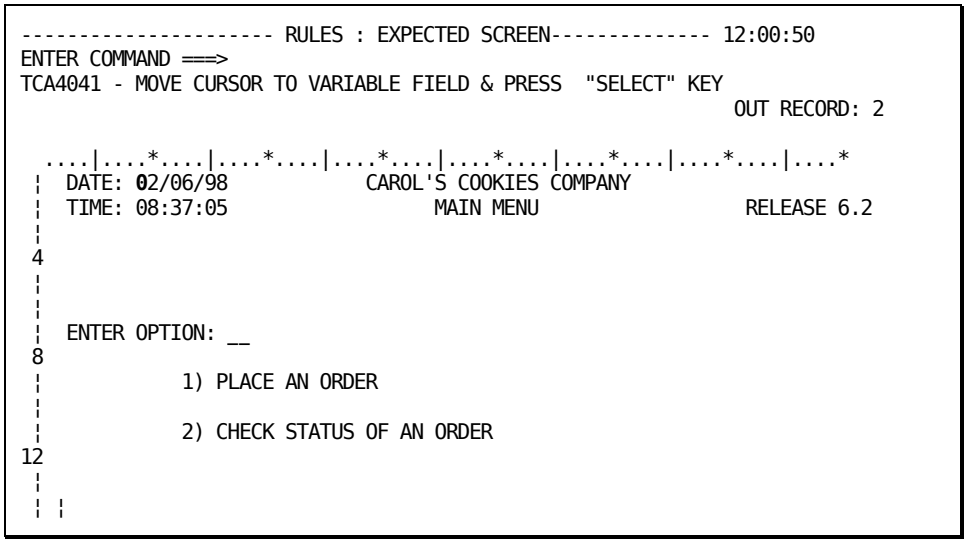
Result:

You will go to the Rules: Expected Screen panel containing your model screen.

Action:

The Date Field: Make the date into a variable:

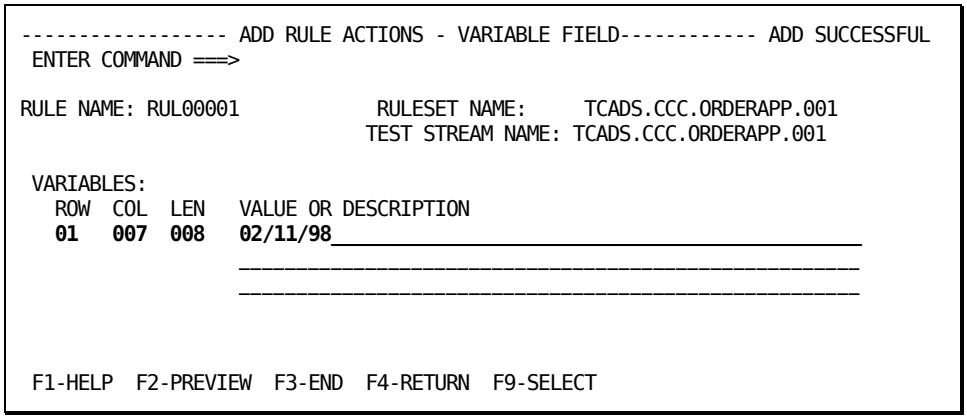
Position the cursor on the first digit of the date, and then press PF9.



Result:

The Add Rule Actions—Variable Field screen will be displayed.

The information for the variable field is entered automatically as shown in the following screen.



Action:

Press PF3.

Result:

You return to the Add Rule - Rules Actions screen.

Action:

The Time Field: To make the Time field variable, select option 3 from the Add Rules—Rule Actions menu.

Type 3 in the command line and press Enter.

```
----------------------ADD RULES - RULE ACTIONS---------------
 ENTER COMMAND ===> 3

RULE NAME: RUL00001           RULESET NAME:     TCADS.CCC.ORDERAPP.001
                             TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

DESCRIPTION  ===> RELEASE 6.2 TO 6.3 CHANGES
             ===>

       1  FIELD RECOGNITION                  11  GENERATE FIELD VALUE
       2  SCREEN RECOGNITION                 12  INSERT SCREENS
       3  VARIABLE FIELD                     13  DELETE SCREENS
       4  DELETE FIELD                       14  CUT SCREEN FIELD
       5  MOVE FIELD                         15  PASTE SCREEN FIELD
       6  CHANGE FIELD VALUE                 20  USERID LOGGING
       7  NEW FIELD                          21  TERMINAL ID LOGGING
       8  CHANGE AID KEY                     22  TRANSACTION ID LOGGING
       9  CHANGE CURSOR LOCATION



  F1-HELP      F2-PREVIEW      F3-END      F4-RETURN      F9-SELECT
```

Result:

You will go to the Rules: Expected Screen panel containing your model screen.

Action:

Position the cursor on the first digit of the time, and then press PF9.

```
---------------------- RULES : EXPECTED SCREEN-------------- 12:00:50
ENTER COMMAND ===>
TCA4041 - MOVE CURSOR TO VARIABLE FIELD & PRESS  "SELECT" KEY
                                                              RECORD: 2

  ....|....*....|....*....|....*....|....*....|....*....|....*....|....*....|....*
 ¦  DATE: 02/06/98           CAROL'S COOKIES COMPANY
 ¦  TIME: 08:37:05                MAIN MENU                 RELEASE 6.2
 ¦
 4
 ¦
 ¦
 ¦  ENTER OPTION: __
 8
 ¦            1) PLACE AN ORDER
 ¦
 ¦            2) CHECK STATUS OF AN ORDER
12
 ¦
 ¦
 ¦
```

Result:

The Add Rule Actions—Variable Field screen displays.

The information for the variable field is entered automatically as shown in the following screen.

```
---------------- ADD RULE ACTIONS - VARIABLE FIELD-------------  ADD SUCCESSFUL
 ENTER COMMAND ===>

RULE NAME: RUL00001              RULESET NAME: TCADS.CCC.ORDERAPP.001
                            TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

 VARIABLES:
   ROW  COL  LEN   VALUE OR DESCRIPTION
   02   007  008   14:06:31_____________________________________________
                   ____________________________________________________
                   ____________________________________________________


 F1-HELP  F2-PREVIEW  F3-END  F4-RETURN  F9-SELECT
```

Action:

Press PF3.

Result:

You return to the Add Rule—Rules Actions screen. You have now completed the field changes required for the Release Level rule.

Check the Summary Information

Check the rules you have created.

Action:

Press PF3 twice.

Result:

The Maintain Rules menu is displayed.

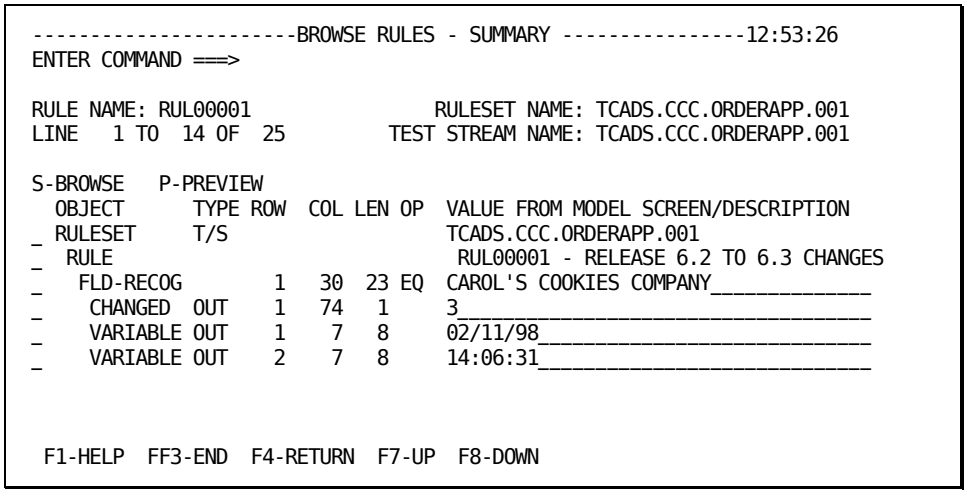
Action:

Type B in the command line.

```
------------------------- MAINTAIN RULES ----------------------12:05:38
ENTER COMMAND ===> B

   A  ADD A RULESET                C  COPY A RULESET
   B  BROWSE A RULESET             D  DELETE A RULESET
   E  EDIT A RULESET               R  RENAME A RULESET

RULESET NAME:

  DDNAME      ===> TCADS
  APPLICATION ===> ccc             (LEAVE APPLICATION, MEMBER,
  MEMBER      ===> orderapp         OR VERSION BLANK AND PRESS
  VERSION     ===> 001              ENTER FOR A SELECTION LIST)









 F1-HELP      F3-END       F4-RETURN
```

Result:

The Browse Rules—Summary panel is displayed as shown next.



```
----------------------BROWSE RULES - SUMMARY ----------------12:53:26
ENTER COMMAND ===>

RULE NAME: RUL00001              RULESET NAME: TCADS.CCC.ORDERAPP.001
LINE   1 TO  14 OF  25         TEST STREAM NAME: TCADS.CCC.ORDERAPP.001

S-BROWSE   P-PREVIEW
  OBJECT      TYPE ROW  COL LEN OP  VALUE FROM MODEL SCREEN/DESCRIPTION
_ RULESET     T/S                   TCADS.CCC.ORDERAPP.001
_  RULE                              RUL00001 - RELEASE 6.2 TO 6.3 CHANGES
_   FLD-RECOG        1   30  23 EQ  CAROL'S COOKIES COMPANY_____________
_    CHANGED  OUT    1   74   1     3_____________________________________
_    VARIABLE OUT    1    7   8     02/11/98______________________________
_    VARIABLE OUT    2    7   8     14:06:31______________________________


 F1-HELP  FF3-END  F4-RETURN  F7-UP  F8-DOWN
```

From this panel you can display or change any of the rule actions, rules, or rulesets that are defined for this test stream.

Action:

Press PF3 twice.

Result:

You return to the Primary Options menu.

Step 3. Exit CA Verify for CICS

In Steps 1 through 3, you created a set of rules to be used with the Carol's Cookies Company Release 6.2 test stream. Now you need to exit CA Verify for CICS in preparation for testing Release 6.3.

Action:

Select option x from the Primary Options Menu and press Enter.

```
VERSION x.x.x -------  CA VERIFY PRIMARY OPTIONS MENU  ----------------13:44:59
ENTER COMMAND ===>  x                                                TERM: 60L2050
                                                                    OPER:
   L  LOG A TEST STREAM

   B  BROWSE A TEST STREAM

   R  RUN A TEST STREAM

   E  EDIT A TEST STREAM

   M  MAINTAIN RULES

   I  INQUIRY/TERMINATION OF FUNCTIONS

   U  UTILITIES

   T  TUTORIAL

   X  EXIT
         *** ENTER "KEYS" COMMAND TO SEE PF KEY ASSIGNMENTS ***


COPYRIGHT © 2012 CA.   ALL RIGHTS RESERVED
```

Result:

You will exit CA Verify for CICS.

This is the end of Part 2 of the demo. You can continue to Part 3 or stop the demo at this time.

Part 3: Test Release 6.3

In Part 3 of this demo, you will run your test stream from Part 1 with the Carol's Cookies Company program Release 6.3. You will also identify any unexpected discrepancies between the output screens from your test stream (Release 6.2) and the Release 6.3 output.

Note: If a lot of time has elapsed since you have completed Parts 1 and 2 of this demo, then you should check to see if the Install 6.3 task has been completed. If it has not (or the files are gone), then re-do Part 2 before beginning Part 3.

To do this, follow these tasks:

1. Start the Run session.
2. Examine the mismatch.
3. Terminate the run.
4. Exit CA Verify for CICS.

Steps 1 and 2 involve sub-tasks that must be followed for the step to be completed.

At the end of this section, you will have tested Release 6.3, detected an oversight in the rules created in Part 2, and discovered a bug in this version of the Carol's Cookies Company program.

Step 1. Start the Run Session

To start the run session, complete these tasks:

A. Invoke CA Verify for CICS.

B. Select the Run function from the Primary Options Menu.

C. Specify options for this run.

D. Activate the run.

A. Invoke CA Verify for CICS

Action:

Clear your screen (if it is not already clear). Type xtca and press Enter.

Result:

CA Verify for CICS displays the Primary Options Menu.

B. Select the Run Function

Action:

Type r and press Enter.

```
VERSION x.x.x---------- CA VERIFY PRIMARY OPTIONS MENU  ----------10:29:28
ENTER COMMAND ===> r                                              TERM: L9D6CA
                                                                  OPER: NGN
   L  LOG A TEST STREAM

   B  BROWSE A TEST STREAM

   R  RUN A TEST STREAM

   E  EDIT A TEST STREAM

   M  MAINTAIN RULES

   I  INQUIRY/TERMINATION OF FUNCTIONS

   U  UTILITIES

   T  TUTORIAL

   X  EXIT
          *** ENTER "KEYS" COMMAND TO SEE PF KEY ASSIGNMENTS ***


COPYRIGHT © 2012 CA.   ALL RIGHTS RESERVED
```

Result:

CA Verify for CICS displays the Run Options menu.

C. Specify Run Options

Some of the fields on this menu contain default values. Complete the menu as illustrated next, typing information as needed.

Action:

For Application, type ccc. For Member, type orderapp. Type 001 for Version. Type * for the new output test stream ddname. When you are finished, press Enter.

Since the value for the Process With Rules field is Y, the Run will include the test stream rules you created in Part 2. (It will also include any application and system rulesets you have created).

Note: Typing N tells CA Verify for CICS not to use rulesets when running this test stream. Typing S (select) allows you to choose which of the three types of rulesets (test stream, application, or system) you want applied to this run.

```
------------------------------  RUN OPTIONS  ------------------------12:54:57
ENTER COMMAND ===>

ENTER INPUT TEST STREAM NAME:
  DDNAME      ===> TCADS
  APPLICATION ===> ccc            (LEAVE APPLICATION, MEMBER,
  MEMBER      ===> orderapp         OR VERSION BLANK AND PRESS
  VERSION     ===> 001              ENTER FOR A SELECTION LIST)

 CREATE NEW OUTPUT TEST STREAM:
  DDNAME      ===> *               ("*" USES NEXT AVAILABLE VERSION)
  APPLICATION ===>
  MEMBER      ===>
  VERSION     ===>

PROCESS WITH RULES         ===> Y       (Y/N OR S-SELECT)
COMPARISON TYPE            ===> S       (S-SCREEN, L-LOGICAL, P-PHYSICAL)
RECORD HISTORY             ===> Y       (Y/N)
  REQUIRE SIGNOFF DATA     ===> N       (Y/N)
SIMULATED USER THINK TIME  ===> NONE    (NONE, NNN% OF ORIGINAL, NN SECONDS)
STATUS INTERVAL            ===> 005     (SECONDS)
CANCEL INTERVAL            ===> 001     (MINUTES)
STOP AT MISMATCHES         ===> Y       (Y/N)

F1-HELP        F3-END       F4-RETURN
```

Result:

The panel will redisplay with the name of the new output test stream filled in with the version as 002. Press Enter to begin the run; the Run Status panel will appear.

D. Activate the Run

When you completed task C and pressed Enter on the Run Options panel, the run was activated. The Run Status panel will briefly appear and looks like this:

```
------------------------------  RUN STATUS  -------------------------11:44:02
                                                                        R3

 INPUT TEST STREAM:     TCADS.CCC.ORDERAPP.001

 OUTPUT TEST STREAM:    TCADS.CCC.ORDERAPP.002

 RULESET(S):
   TESTSTREAM:  TCADS.CCC.ORDERAPP.001
   APPLICATION: NOT USED
   SYSTEM:      NOT USED

 CURRENTLY PROCESSING RECORD:         1
 TOTAL RECORDS TO BE PROCESSED:       12
 ESTIMATED REMAINING RUN TIME: 00:00:36

 STATUS INTERVAL:  5 SECONDS
 CANCEL INTERVAL:  1 MINUTE

 DESCRIPTION: CAROL'S COOKIES COMPANY DEMO PROGRAM
```

Note: The Run Status panel displays until a mismatch is found or until the run is complete.

Step 2. Examine the Mismatch

In this step, you will examine the mismatch found by CA Verify for CICS and terminate the run. To perform this step, you must complete these tasks:

A. Examine the mismatch presented in the Run Mismatch Options screen.

B. Review the rulesets that were applied during this run.

A. Examine the Mismatch

The error that stopped the run is displayed on the bottom of the Run Mismatch Options screen. The Total field, which shows the total value of John Smith's cookie order, has changed from 17.90 in the Release 6.2 test stream to 19.90 in the Release 6.3 run. This is not an expected screen change and indicates a programming error in the new release.

The screen also indicates that this is the only mismatch that was found. From the Run Mismatch Options screen, you can display a summary of the rulesets that were applied during the run.

Action:

Type **1** in the command line and press Enter to display the ruleset summary.

```
 --------------------------  RUN MISMATCH OPTIONS  --------------------11:38:45
 ENTER COMMAND ===>  1
6

     1  DISPLAY RULESET SUMMARY              6  ACCEPT ORIGINAL OUTPUT
     2  DISPLAY PREVIOUS INPUT               7  ACCEPT CURRENT OUTPUT
     3  DISPLAY NEXT INTPUT                  8  CHANGE NEXT INPUT
     4  SKIP ORIGINAL OUTPUT                 9  INSERT CURRENT OUTPUT AND INPUT
     5  SKIP ORIGINAL OUTPUT AND INPUT       10 INSERT CURRENT OUTPUT

INPUT:   TCADS.CCC.ORDERAPP.001         CURRENT RECORD:  6
OUTPUT:  TCADS.CCC.ORDERAPP.002

RULESET: TESTSTREAM:  TCADS.CCC.ORDERAPP.001
         APPLICATION: NOT USED
         SYSTEM:      NOT USED

          TYPE     OPERATION          WCC   CURSOR   SIZE    LENGTH   TERMINAL
EXPECTED: OUTPUT   ERASE/WRITE        C2     1   1   24*80      471   A60L2050
CURRENT:  OUTPUT   ERASE/WRITE        C2     1   1   24*80      471   VV02
ONLY UNEQUAL ROW:  20
               .TOTAL:                  . 17.90
               .TOTAL:                  . 19.90
                                          X
F1-HELP     F2-ROTATE     F3-END     F7-UP     F8-DOWN
```

Result:

The Applied Rules panel displays.

CA Verify for CICS lets you to change or add a rule to a ruleset without terminating the run. For instance, if you forgot to account for an expected screen change, you can select option 1 to display the Applied Rules panel. From here, rulesets and individual rules can be browsed, edited, or deleted. After you modify a ruleset and reactivate the run, CA Verify for CICS will repeat the comparison, taking into account your rule specification. CA Verify for CICS also provides other options for dealing with mismatches during a run, including ignoring the mismatch to continue the run, and updating or replacing a screen in the test stream. See the chapter *Run Function* for more information.

B. Review the Rulesets Applied During the Run

The Applied Rules panel lists all of the rulesets that you requested to be included in this run: test stream, application, and system. From this display, you can see that the identification criteria was recognized and the rule actions defined to the RUL00001 rule were applied.

Action:

Press PF2 to rotate the display to Expected, which consists of the originally-logged test stream record with the rules applied; to the current view of the test stream record; and to the original test stream record from the originally-logged test stream. When you are finished, press PF3.

```
---------------------------- APPLIED RULES -------------------------12:47:59
 ENTER COMMAND ===>

 INPUT:   TCADS.CCC.ORDERAPP.001                      CURRENT RECORD:  6
                                                       LINE   1 TO  6 OF  6

 B-BROWSE   E-EDIT
   OBJECT      TYPE ROW  COL LEN OP  VALUE FROM THE MODEL SCREEN/DESCRIPTION
 _ RULESET     T/S                  TCADS.CCC.ORDERAPP.001
 _  RULE                             RUL00001- RELEASE 6.2 TO 6.3 CH
APPLIED>
 _    FLD-RECOG        1   30  23 EQ  CAROL'S COOKIES COMPANY______ <RECOGNIZED>
 _     CHANGED  OUT    1   74   1     3_____________________________ <APPLIED>
 _     VARIABLE OUT    1    7   8     02/11/98______________________ <APPLIED>
 _     VARIABLE OUT    2    7   8     13:39:48______________________ <APPLIED>



  F1-HELP     F2-ROTATE     F3-END     F4-RETURN     F7-UP     F8-DOWN
```

Result:

You return to the Run Mismatch Options screen. From there, you will terminate the run.

Step 3. Terminate the Run

CA Verify for CICS uncovered an error in the Release 6.3 program. You will terminate this run and test Release 6.3 FIX to check that this error has been resolved.

Action:

Press **PF3** to view termination options.

Result:

The Run Termination screen appears.

Action:

Type 2 on the command line. Press Enter. Press PF3 to return to the Primary Options Menu.

```
------------------------------  RUN TERMINATION  ---------------11:35:49
ENTER COMMAND ===> 2                                                  RX

   1  END RUN, SAVE OUTPUT TEST STREAM

   2  CANCEL RUN, DO NOT SAVE OUTPUT TEST STREAM

INPUT TEST STREAM:
  TCADS.CCC.ORDERAPP.001

OUTPUT TEST STREAM:
  TCADS.CCC.ORDERAPP.002

RULESET(S) IN USE:
  TESTSTREAM:  TCADS.CCC.ORDERAPP.001
  APPLICATION: NOT USED
  SYSTEM:      NOT USED


CURRENTLY PROCESSING RECORD: 6


F1-HELP       F3-END                                                Action:
```

Result:

The run will be terminated. The Primary Options menu displays.

Step 4. Exit CA Verify for CICS

You need to exit CA Verify for CICS before testing Release 6.3 FIX.

Action:

Select option x from the Primary Options Menu and press Enter.

```
VERSION x.x.x -------  CA VERIFY PRIMARY OPTIONS MENU  ----------------13:44:59
ENTER COMMAND ===>  x                                                TERM: 60L2050
                                                                    OPER:
   L  LOG A TEST STREAM

   B  BROWSE A TEST STREAM

   R  RUN A TEST STREAM

   E  EDIT A TEST STREAM

   M  MAINTAIN RULES

   I  INQUIRY/TERMINATION OF FUNCTIONS

   U  UTILITIES

   T  TUTORIAL

   X  EXIT
         *** ENTER "KEYS" COMMAND TO SEE PF KEY ASSIGNMENTS ***


COPYRIGHT © 2012 CA.   ALL RIGHTS RESERVED
```

Result:

You will exit CA Verify for CICS.

Note: This is the end of Part 3 of the demo. You can continue to Part 4 or stop the demo at this time.

Part 4: Test the Debugged Version of Release 6.3

As a result of the run, you know Release 6.3 has an error that must be fixed. You do not actually need to debug Release 6.3 for this demo. A fixed version has been supplied, called 6.3 FIX. In Part 4, you will run the test stream on this version to check that the problems detected by CA Verify for CICS in the last run have been taken care of.

Follow these tasks:

1. Install Release 6.3 FIX.
2. Start the run function.
3. Review the run completion statistics.

Step 1. Install Release 6.3 FIX

Action:

Type install 6.3 fix on a clear screen.

Result:

This message displays:

APPLIED 6.3 FIX

Step 2. Start the Run Session

The fix has now been installed to correct the bug that you found in Part 3. Run the test stream again to make sure that the problem has been solved.

To run the test stream, complete these tasks:

A. Invoke CA Verify for CICS.

B. Select the Run function from the Primary Options Menu.

C. Specify options for this run session and activate the run.

D. Review the Run Statistics and exit CA Verify for CICS.

A. Invoke CA Verify for CICS

Action:

Clear your screen. Type xtca and press Enter.

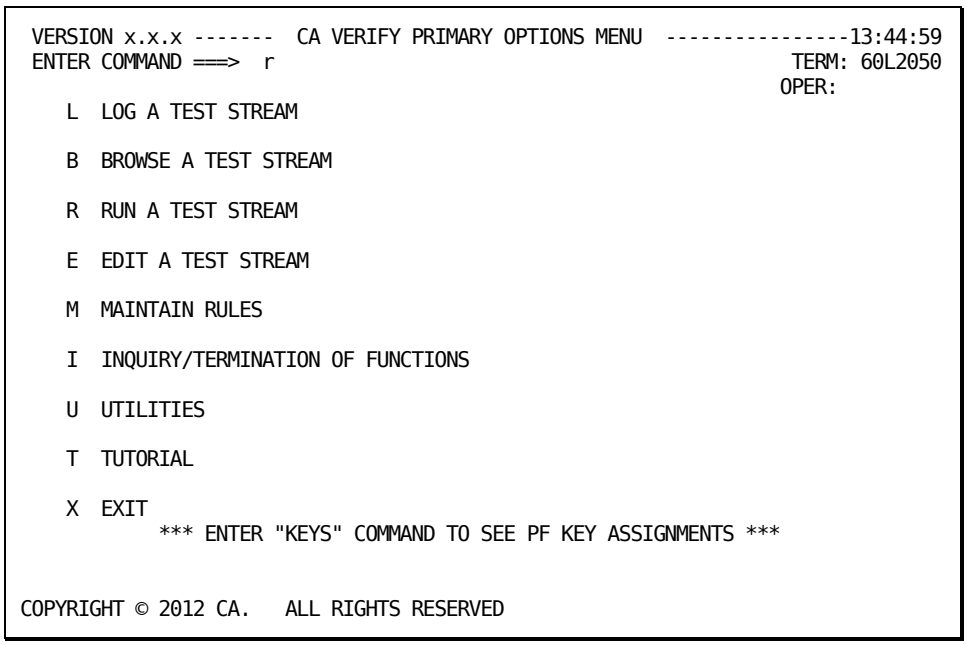
Result:

CA Verify for CICS displays the Primary Options Menu.

B. Select the Run Function

Action:

Type r in the command line and press Enter.



```
VERSION x.x.x -------  CA VERIFY PRIMARY OPTIONS MENU  ----------------13:44:59
ENTER COMMAND ===>  r                                                 TERM: 60L2050
                                                                     OPER:
   L  LOG A TEST STREAM

   B  BROWSE A TEST STREAM

   R  RUN A TEST STREAM

   E  EDIT A TEST STREAM

   M  MAINTAIN RULES

   I  INQUIRY/TERMINATION OF FUNCTIONS

   U  UTILITIES

   T  TUTORIAL

   X  EXIT
         *** ENTER "KEYS" COMMAND TO SEE PF KEY ASSIGNMENTS ***


COPYRIGHT © 2012 CA.   ALL RIGHTS RESERVED
```

Result:

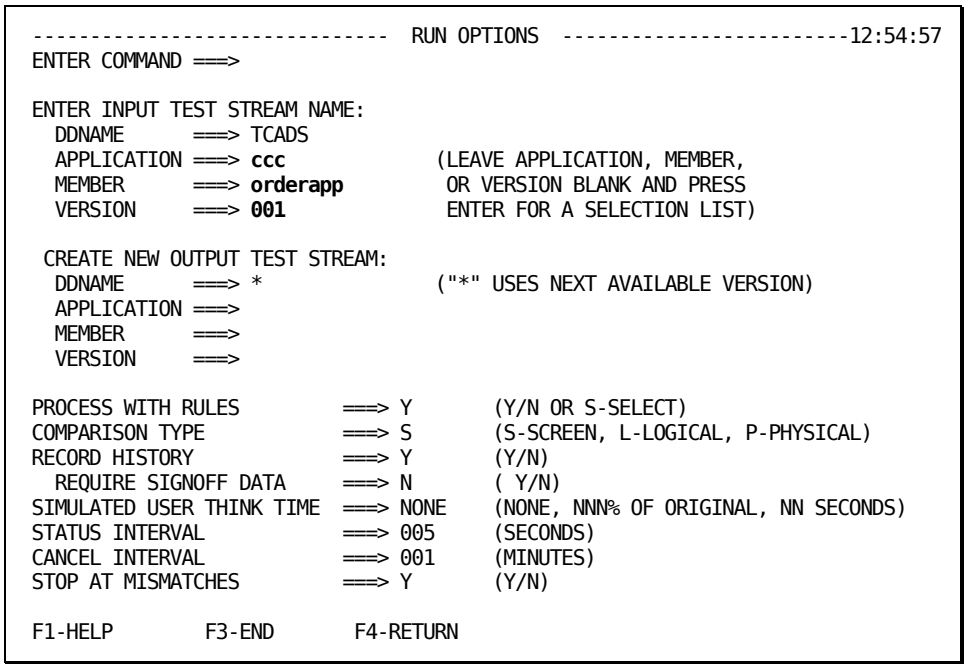
The Run Options menu will be displayed.

C. Specify Run Options and Activate the Run

Some of the fields on this menu contain default values. Complete the menu as illustrated next, typing information as needed.

Action:

For Application, type ccc. For Member, type orderapp. Type ***** in the DDname field to have CA Verify for CICS generate a name for the new output test stream. Press Enter.



Result:

The panel will redisplay with the name of the new output test stream filled in with the version as 002. Press Enter to begin the run; the Run Status panel will appear.

When you completed task C and pressed Enter on the Run Options panel, the run was activated. The Run Status screen will appear as shown next.

```
------------------------------  RUN STATUS  -------------------------11:44:02
                                                                          R3

 INPUT TEST STREAM:     TCADS.CCC.ORDERAPP.001

 OUTPUT TEST STREAM:    TCADS.CCC.ORDERAPP.002

 RULESET(S):
   TESTSTREAM:  TCADS.CCC.ORDERAPP.001
   APPLICATION: NOT USED
   SYSTEM:      NOT USED

 CURRENTLY PROCESSING RECORD:         1

 ESTIMATED REMAINING RUN TIME: 00:00:36

 STATUS INTERVAL:  5 SECONDS
 CANCEL INTERVAL:  1 MINUTE

 DESCRIPTION: CAROL'S COOKIES COMPANY DEMO PROGRAM

F1-HELP     F3-END     ENTER-CONTINUE
```

This screen will be displayed until a mismatch is found or until the run is complete. Since you have corrected the problem, no mismatches will be found and the Run Completion panel will be displayed.

D. Review the Run Statistics

The Run Completion panel provides information on the run that has just completed.

```
TCADS.CCC.ORDERAPP.001 -----------  RUN COMPLETION  -------------------13:02:28
 ENTER COMMAND ===>                                                        R8

 OUTPUT SCREENS COMPARED:        6
 PHYSICALLY EQUAL SCREENS:       1
 LOGICALLY EQUAL SCREENS:        6

 OUTPUT TEST STREAM: TCADS.CCC.ORDERAPP.002

 RULESET(S) USED:
   TESTSTREAM:  TCADS.CCC.ORDERAPP.001
   APPLICATION: NOT USED
   SYSTEM:      NOT USED


                            IN:        OUT:
 EQUAL:                      6           1
 EQUIVALENT:                             5
 IGNORED:                                0
 ACCEPTED:                               0
 INSERTED:                   0           0
 DELETED:                    0           0
 CHANGED:                    0
 NOT RUN:                    0           0


  F1-HELP  F3-END   F4-RETURN
```

Exit the Demo

You have now finished testing the Carol's Cookies Company demo program. With CA Verify for CICS, you were able to:

- Detect program errors
- Ignore valid differences
- Fix a program bug and retest to make sure the error was corrected

If you want, you can print the test stream you just created for documentation purposes. See the "Batch Functions" chapter for directions.

Action:

Press PF3 twice.

Result:

CA Verify for CICS displays the Primary Options Menu.

Now you can either exit from CA Verify for CICS, use the test streams you created to learn more about CA Verify for CICS's other features, or continue on to the Rules Primer to learn more about using the Rules function

Index

A

B

C

D

E

F

H

I

K

L

M

N

O

P

R

S

T

U

V

W